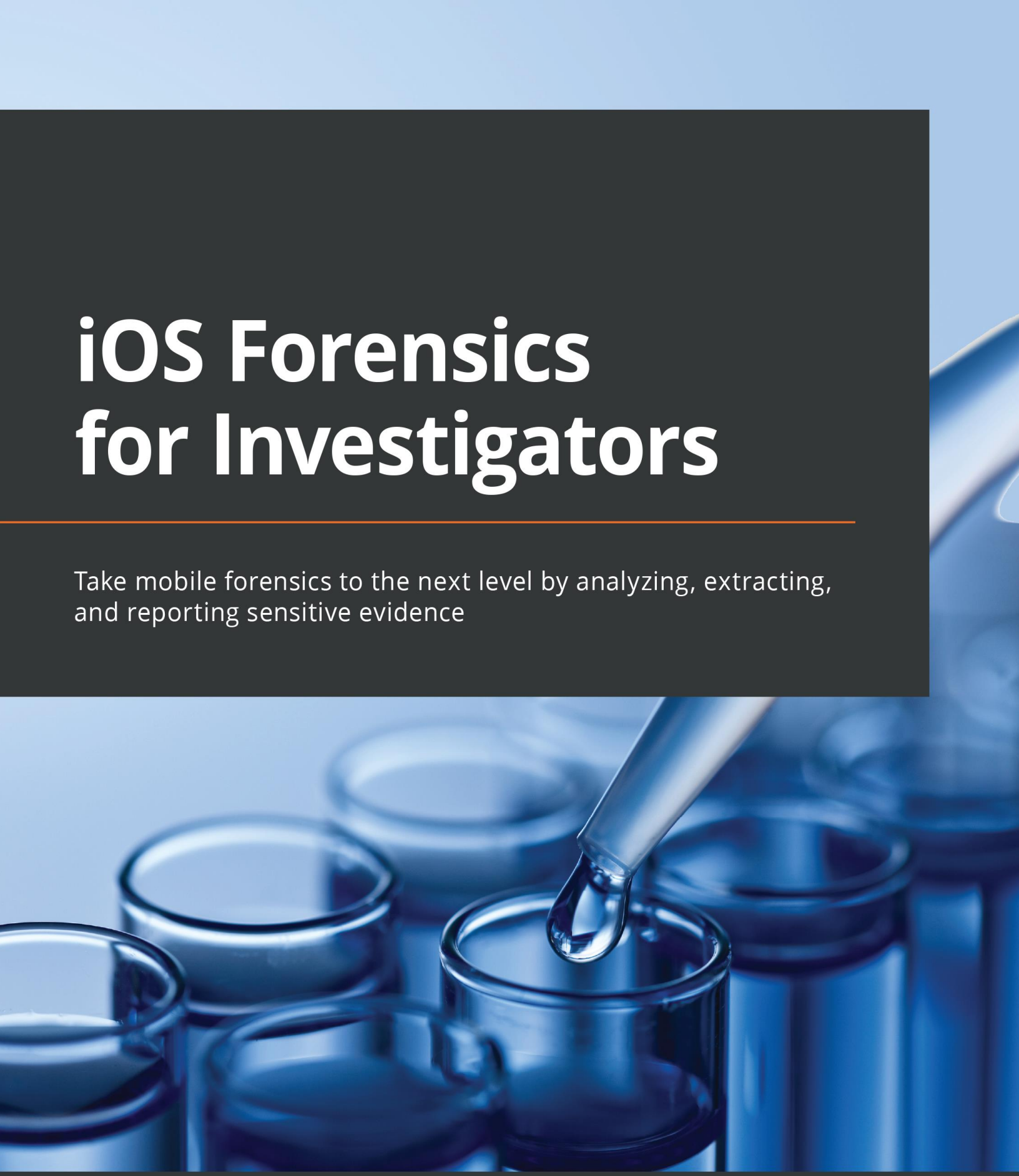
iOS Forensics for Investigators

Take mobile forensics to the next level by analyzing, extracting, and reporting sensitive evidence



Gianluca Tiepolo

Chapter 1

Images

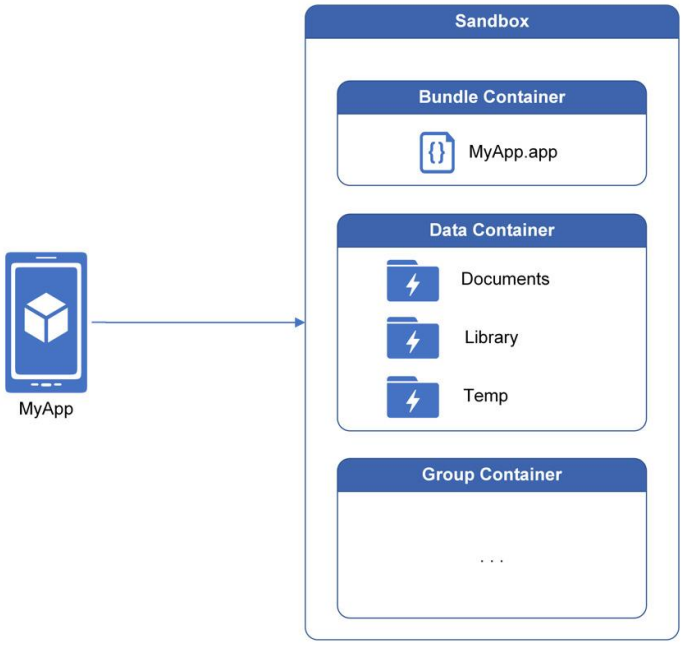


Figure 1.1 – A representation of application containers

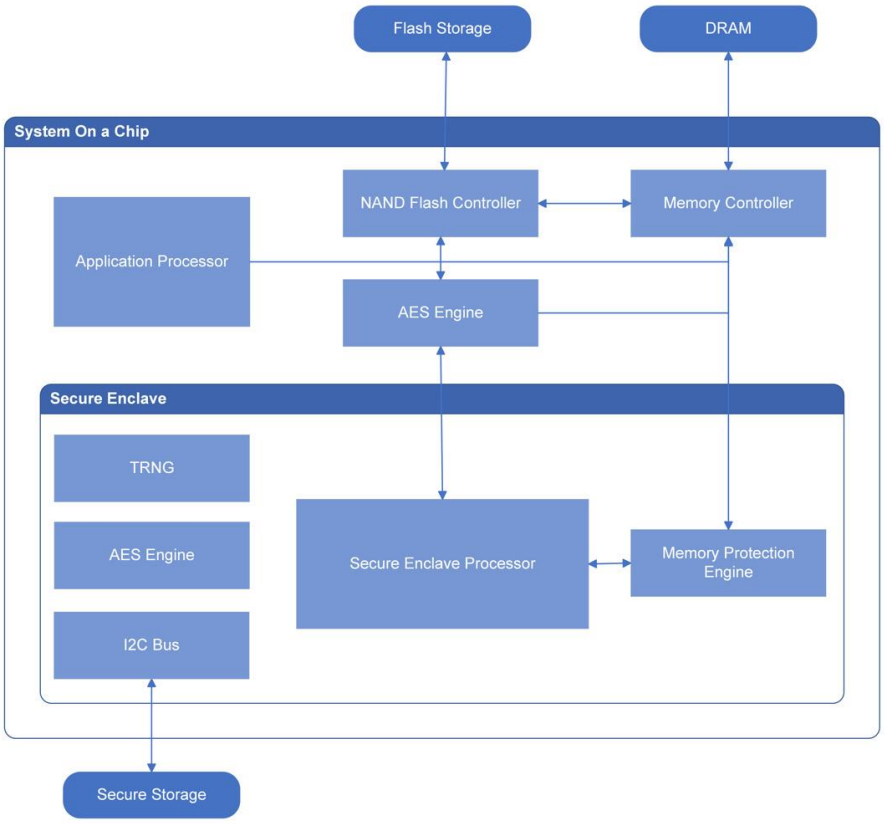


Figure 1.2 – Secure Enclave components

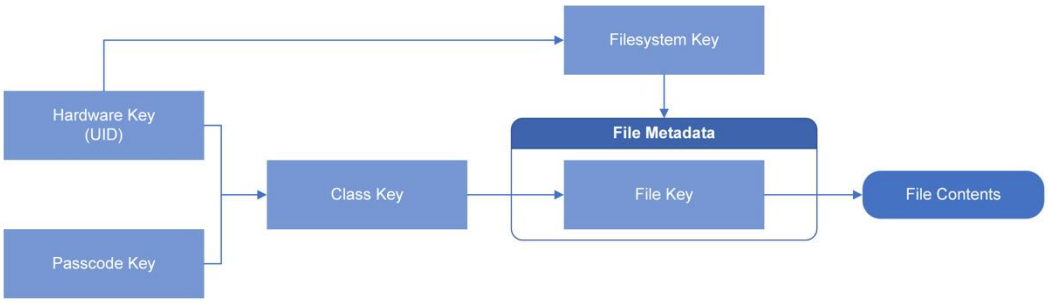


Figure 1.3 – Cryptographic keys used for iOS data protection

Tables

	Logical	Filesystem	Physical
Calls, calendars, contacts, and messages	✓	✓	✓
User content (photos, videos, notes, and so on)	✓	✓	P
Keychain	Limited	✓	✓
Third-party app data	Limited	✓	✓
Apple Pay	-	✓	✓
Location data (radio cells, GPS fixes, and so on)	-	✓	✓
Mail	-	✓	✓
User activity logs	-	✓	✓
System logs	-	✓	✓
Deleted data (unallocated space)	-	-	✓
Deleted data (SQLite databases)	Limited	✓	✓

Table 1.1 – Comparison of data that can be extracted with different acquisition methods

Code and Commands

Code 1.1 – property list in XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

    <dict>

        <key>UUID</key>

        <string>3bdd52c7-ee36-4689-8517-
c5fed2c98s5</string>

        <key>ClientID</key>

        <string>3bdd52c7-ee36-4689-8517-
c5fed2c98s5</string>
```

```
        <key>ClientEnabled</key>
        <false/>
    </dict>
</plist>
```

Chapter 2

Images

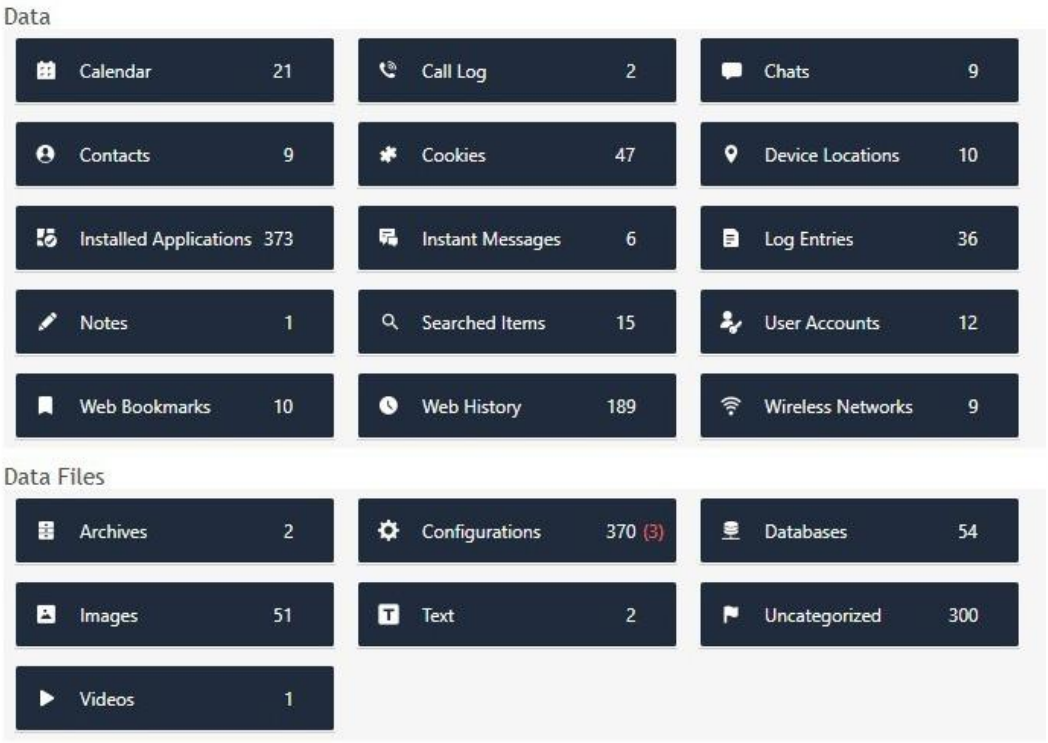


Figure 2.1 – Data acquired through a logical acquisition

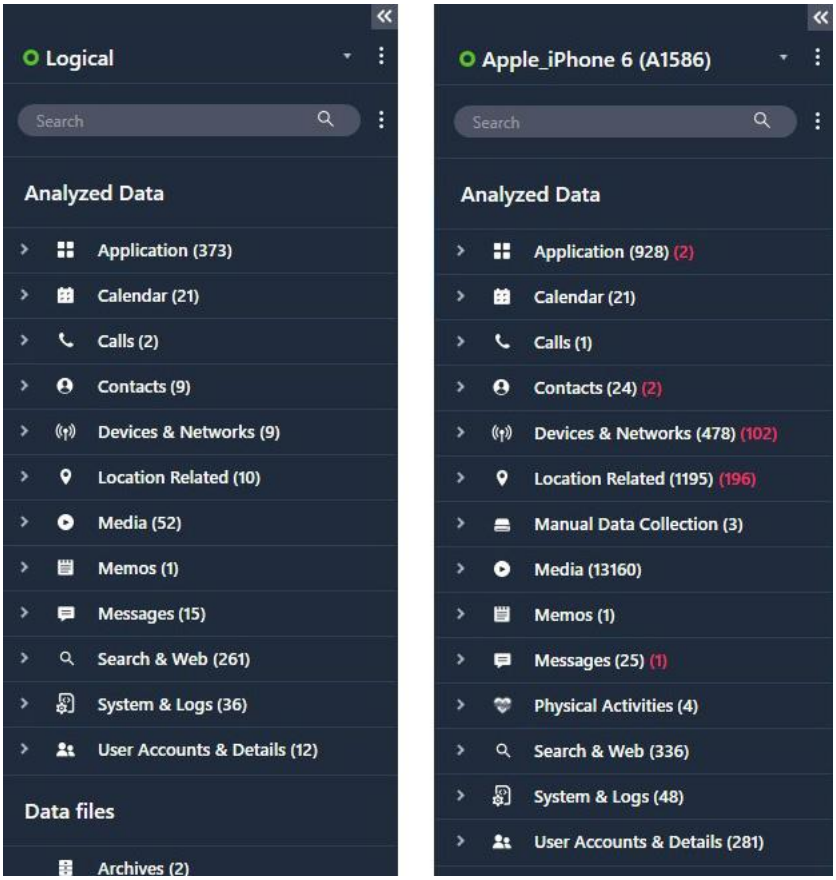


Figure 2.2 – Comparison of data extracted from a logical and a filesystem acquisition

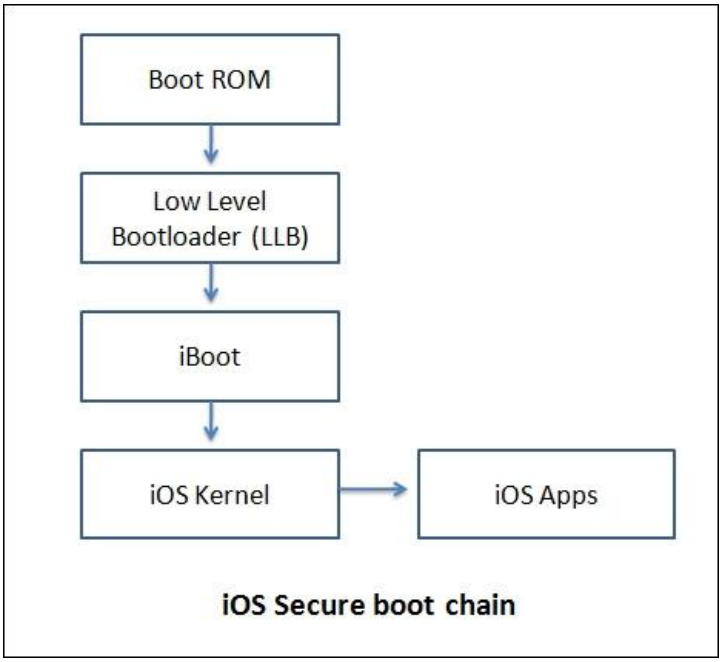


Figure 2.3 – iOS secure boot chain

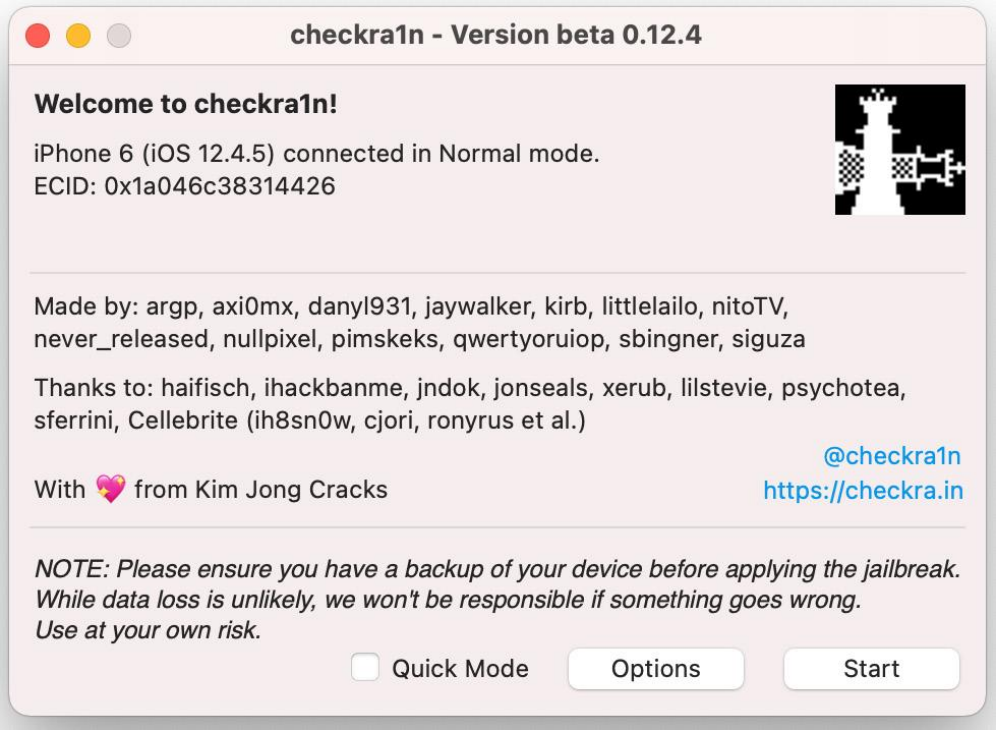


Figure 2.4 – The main check-rain screen

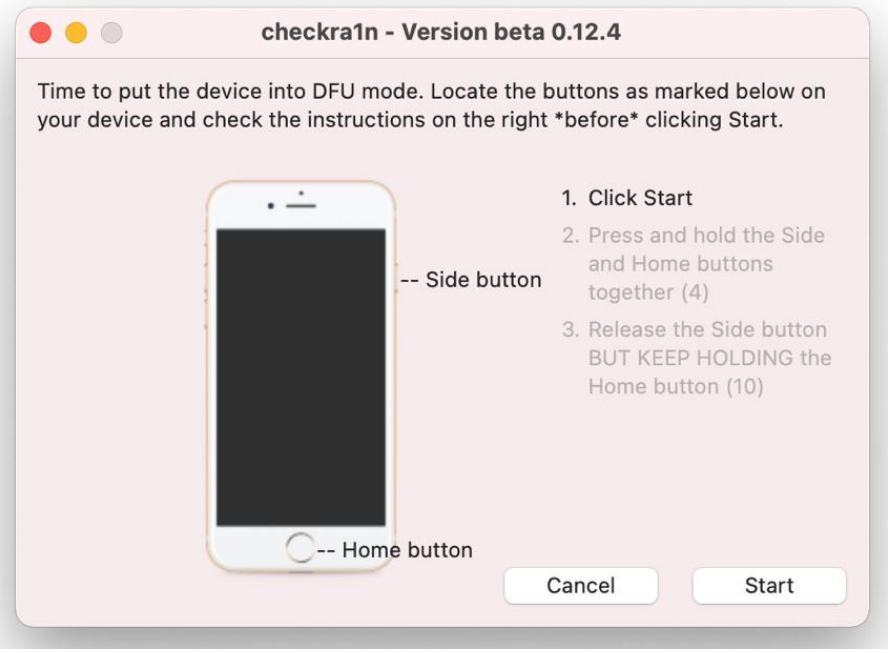


Figure 2.5 – Follow the instructions to put the device in DFU mode

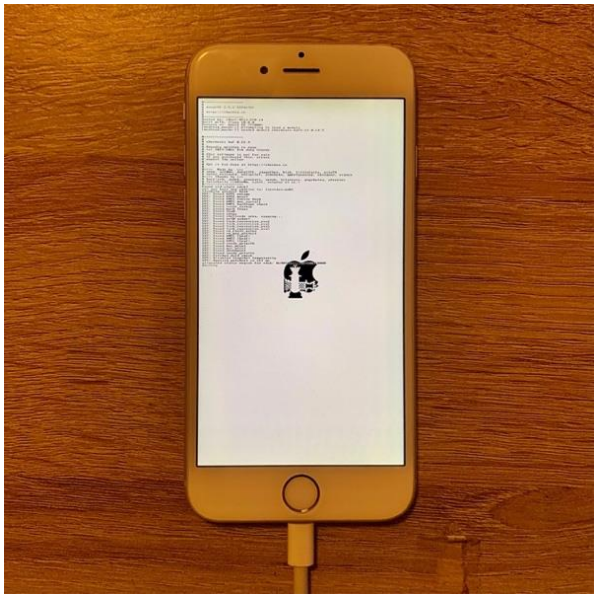


Figure 2.6 – The device booting PongoOS

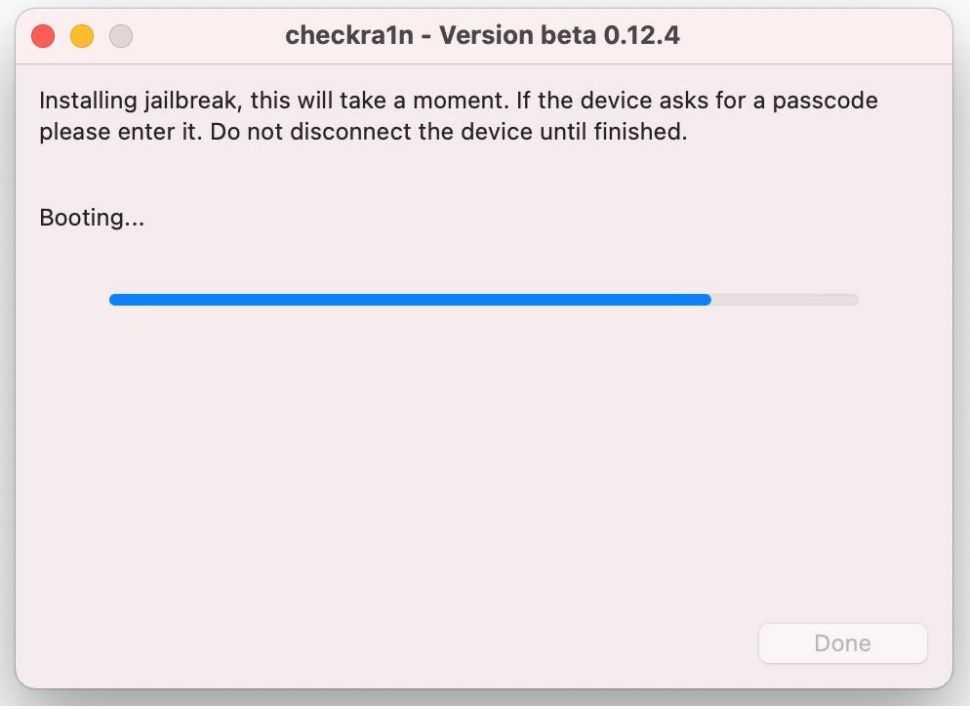


Figure 2.7 – Enter the passcode when requested

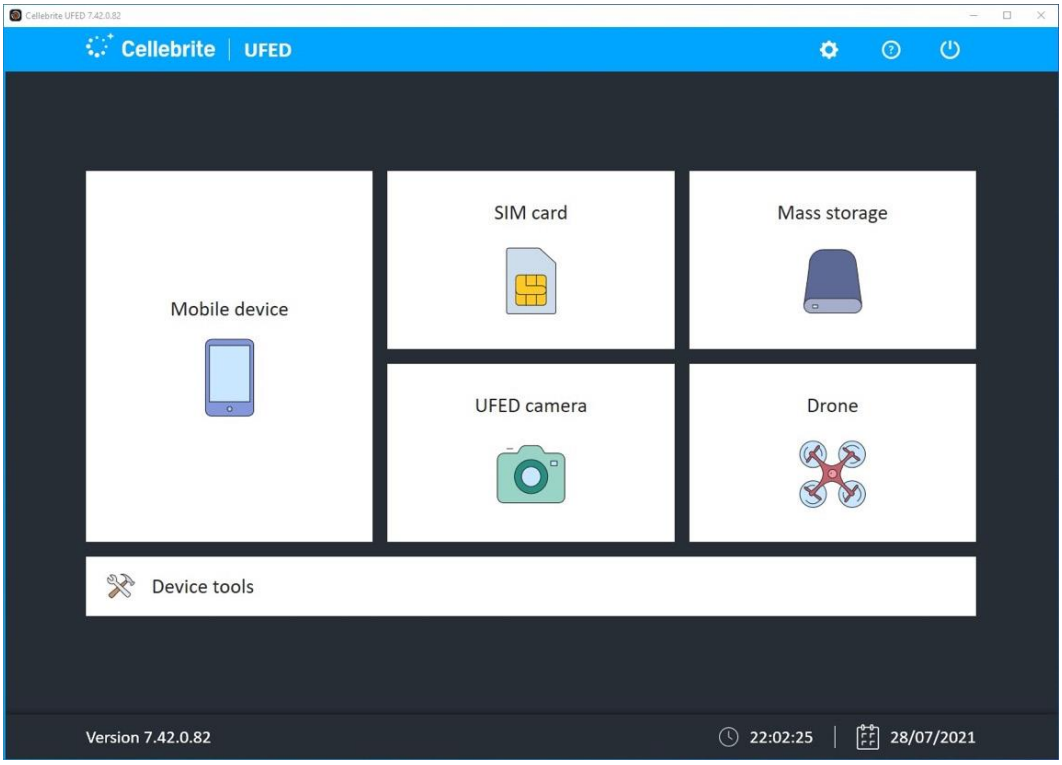


Figure 2.8 – The home screen of Cellebrite UFED

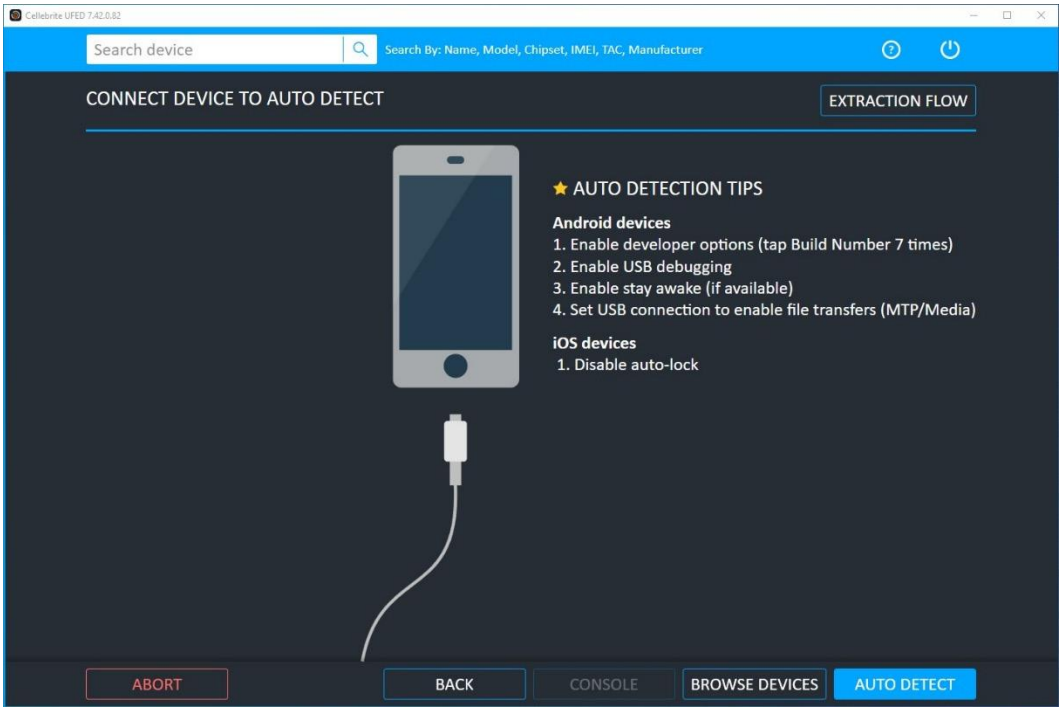


Figure 2.9 – Choose the device or let UFED auto-detect it

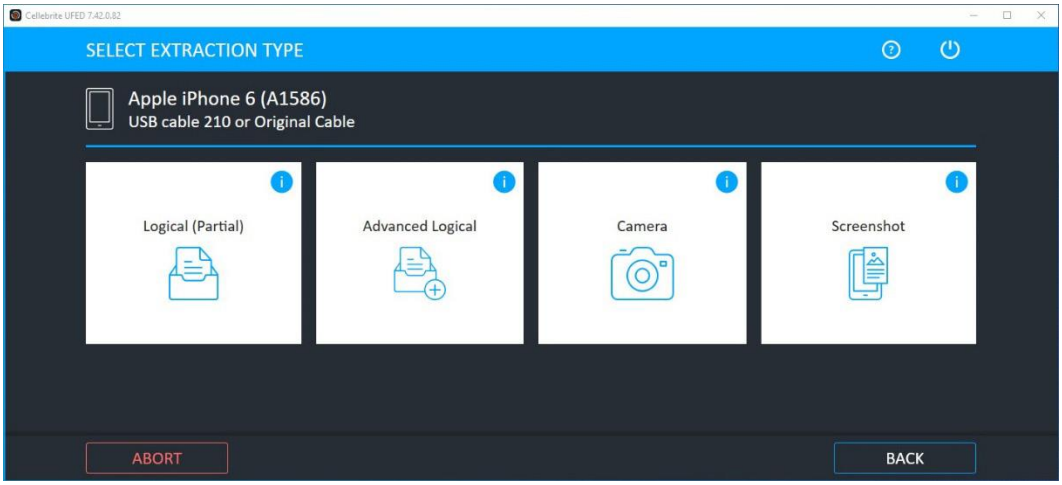


Figure 2.10 – Choose the type of acquisition

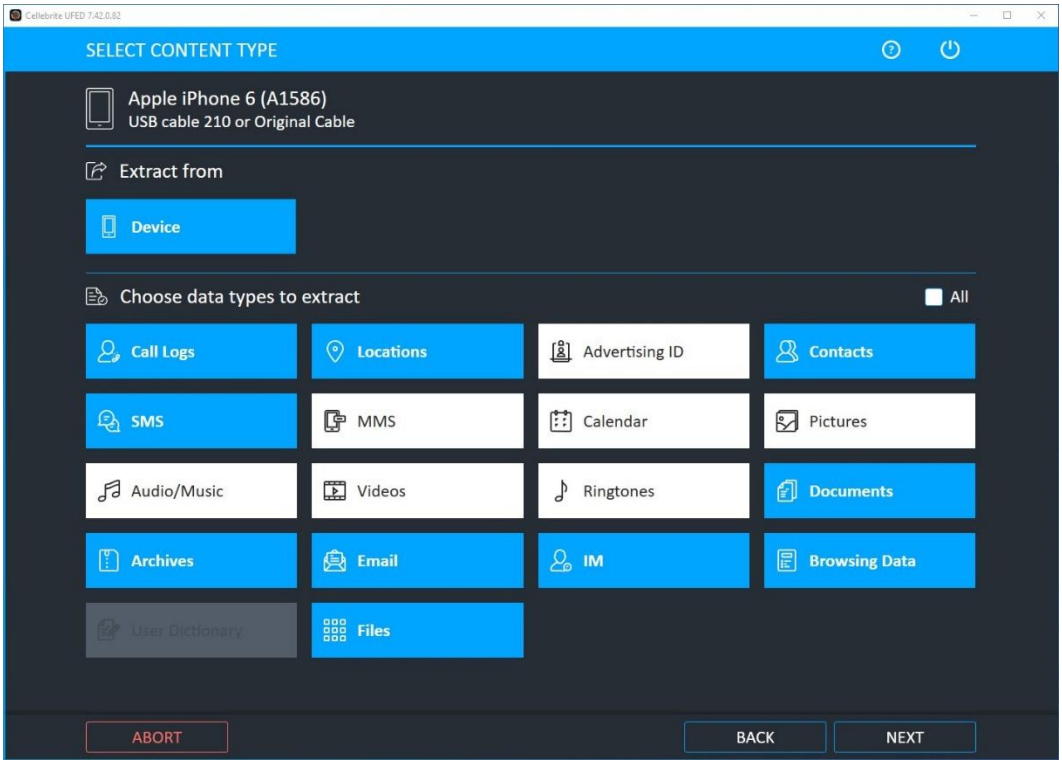


Figure 2.11 – Choose what artifacts should be extracted

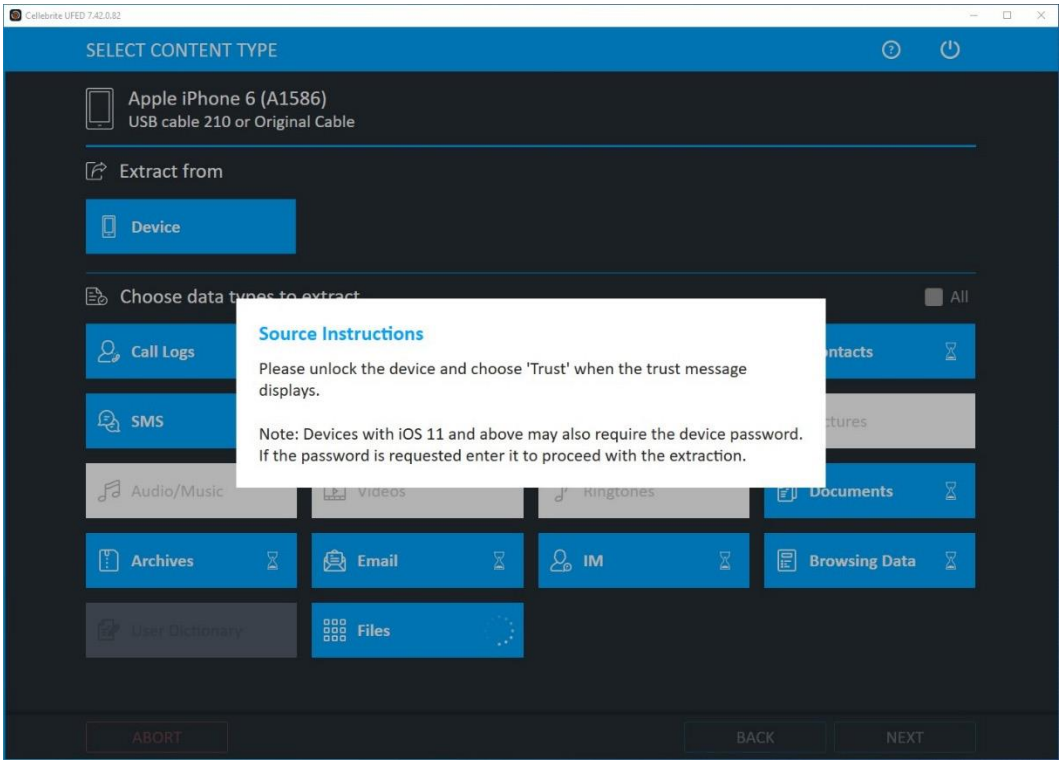


Figure 2.12 – Establish pairing between the workstation and the device

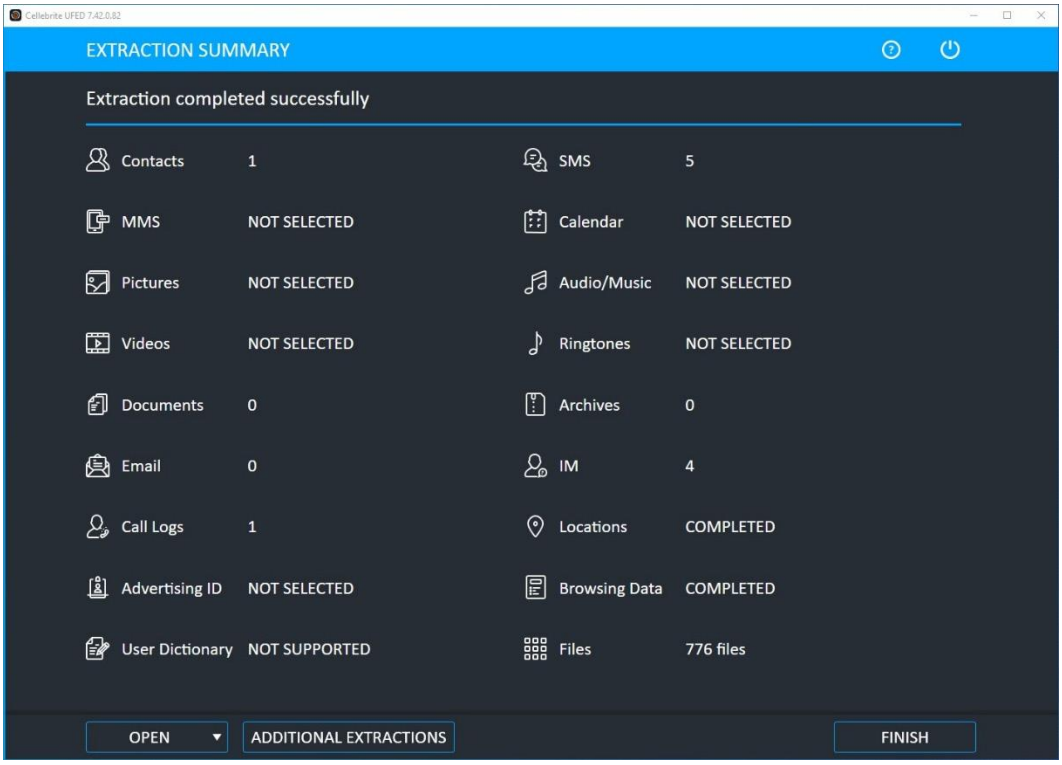


Figure 2.13 – The extraction summary displays what data was extracted

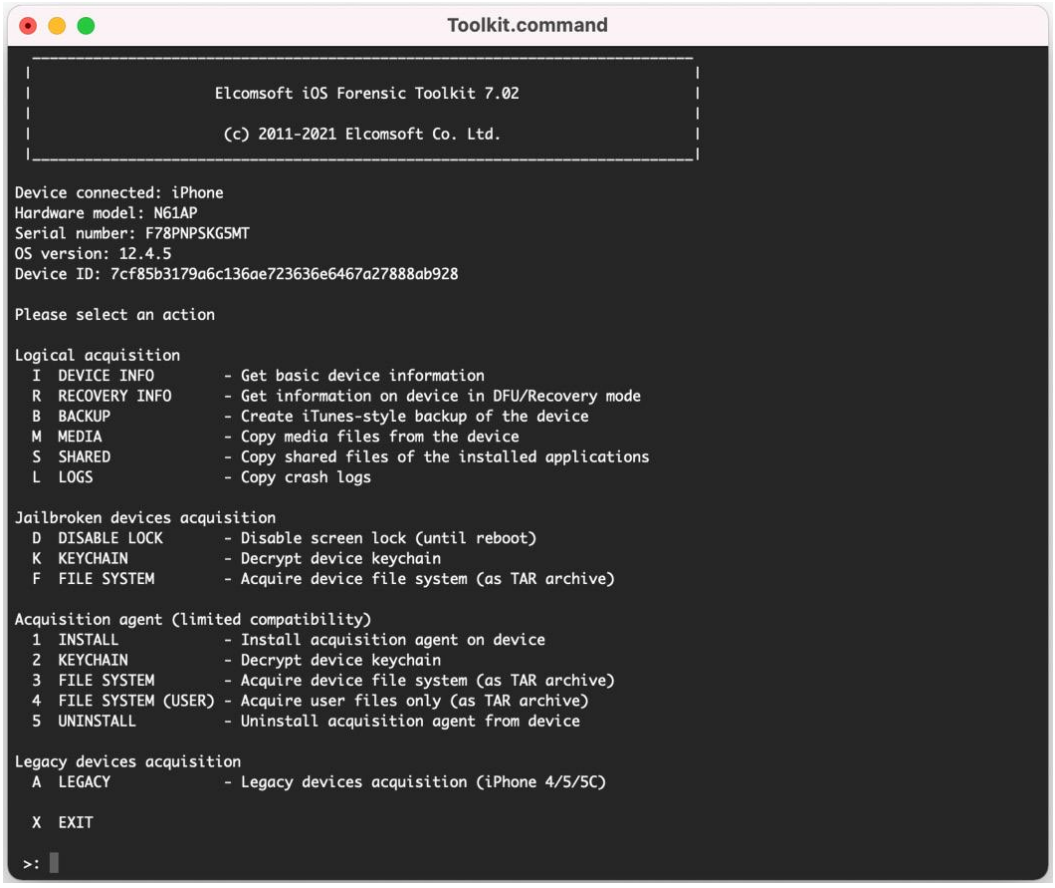


Figure 2.14 – The home screen of Elcomsoft iOS Forensic Toolkit

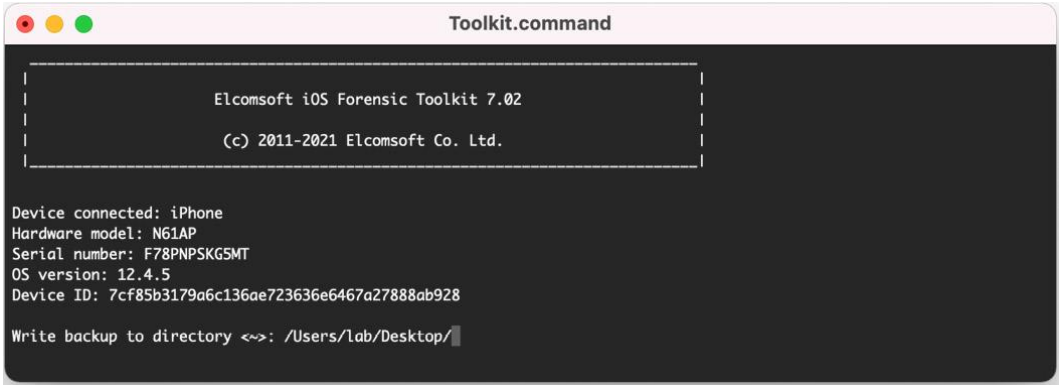


Figure 2.15 – Choose where the backup should be stored

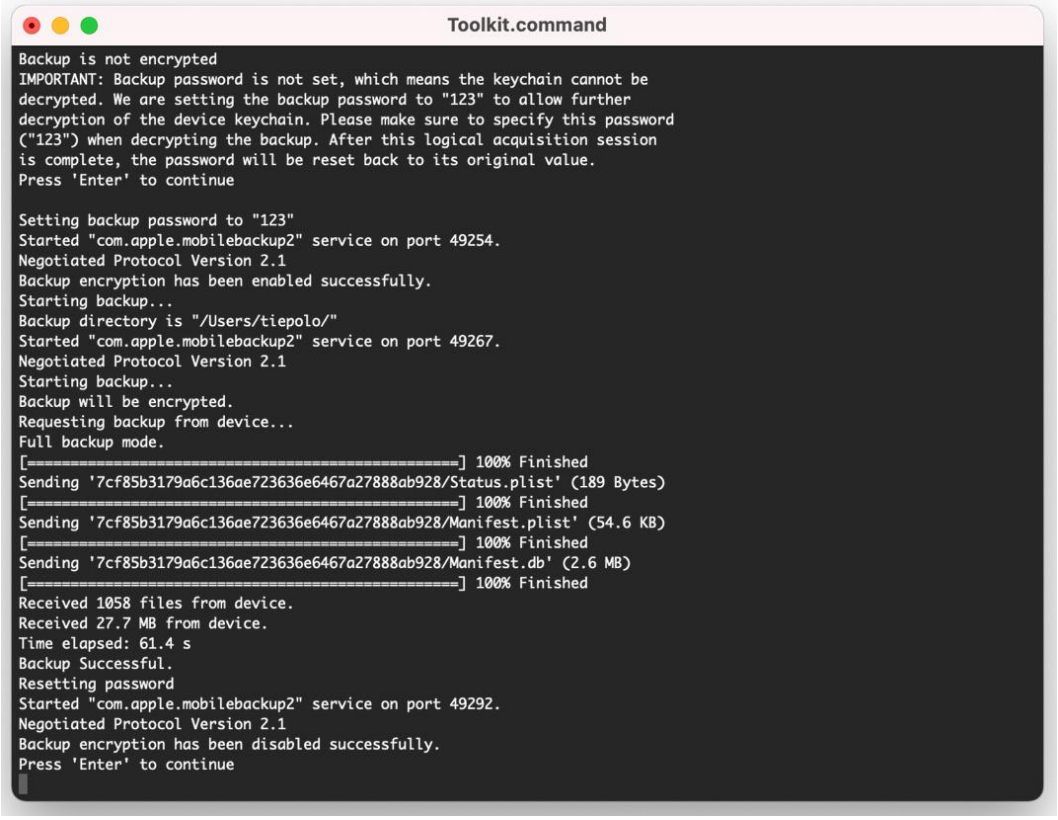


Figure 2.16 – The toolkit will show progress and debug information on screen

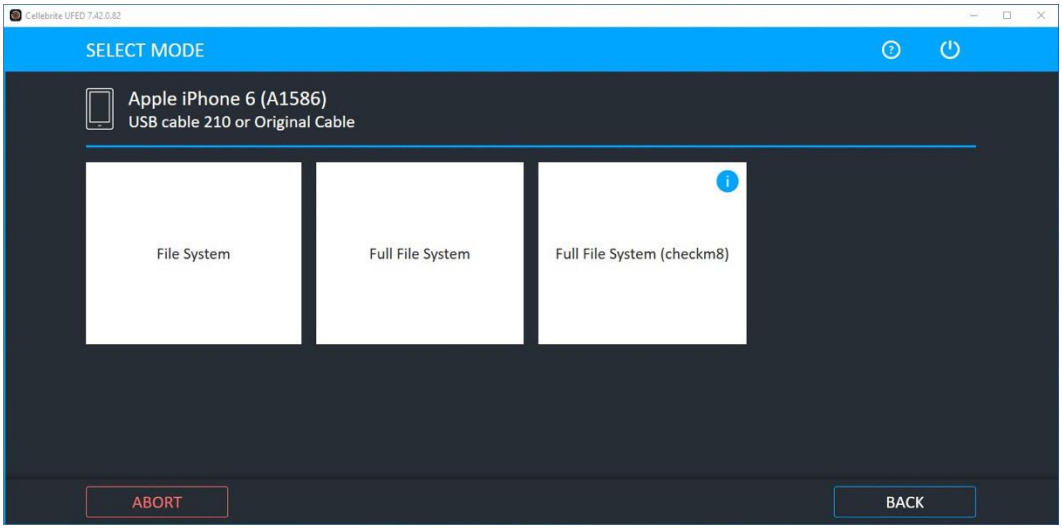


Figure 2.17 – Choose the Full File System (checkmate) option

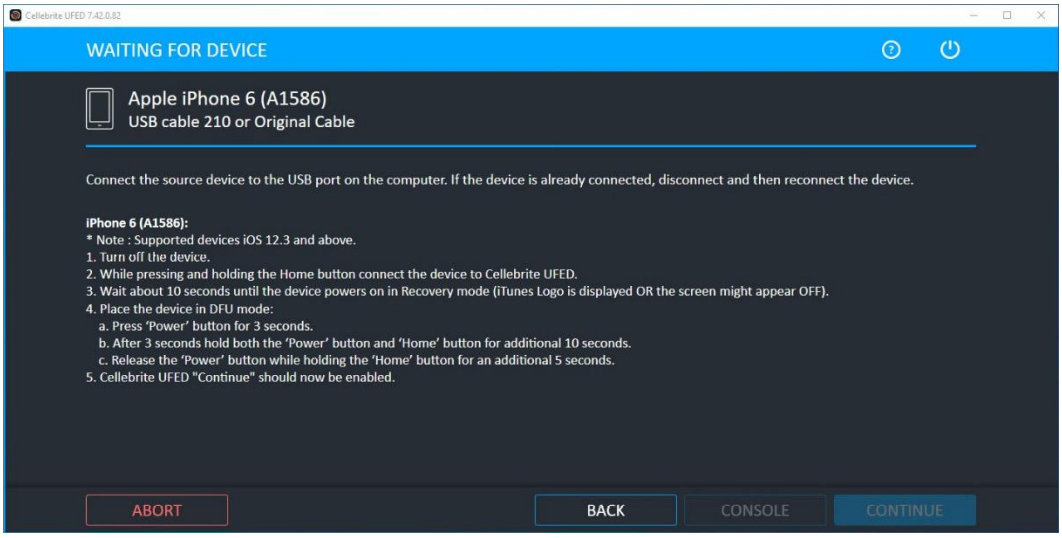


Figure 2.18 – UFED will display instructions on how to enter Recovery and DFU modes

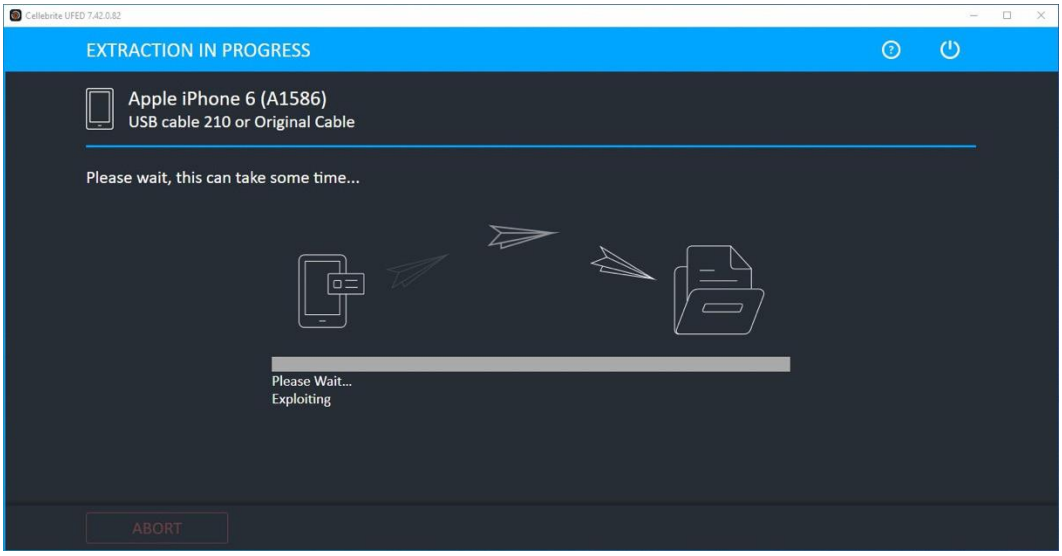


Figure 2.19 – Exploiting the device could take a few minutes

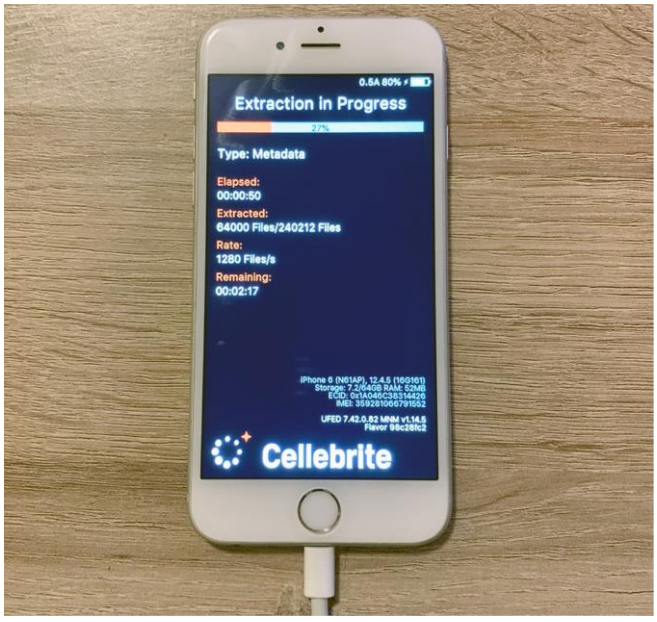


Figure 2.20 – Acquisition in process

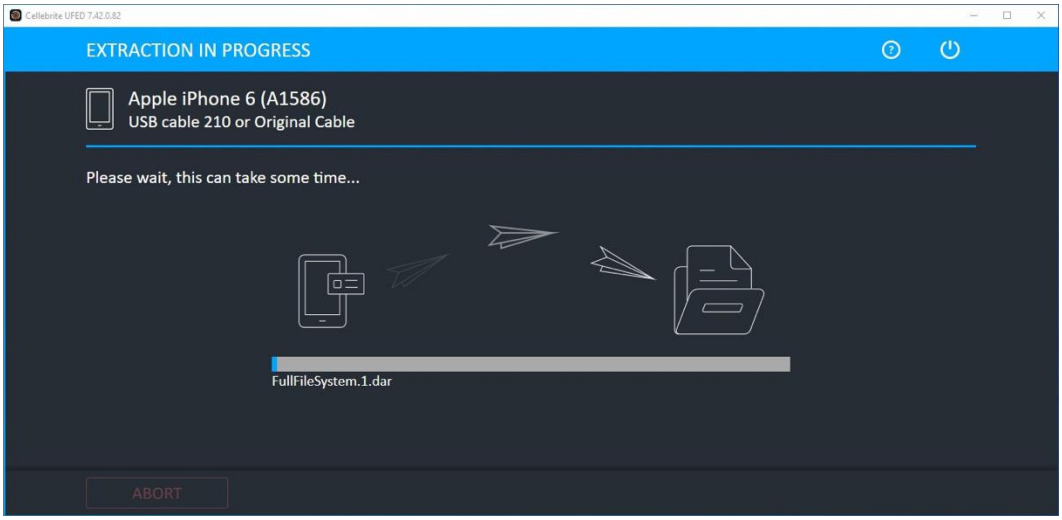


Figure 2.21 – The extraction will be archived in a DAR file

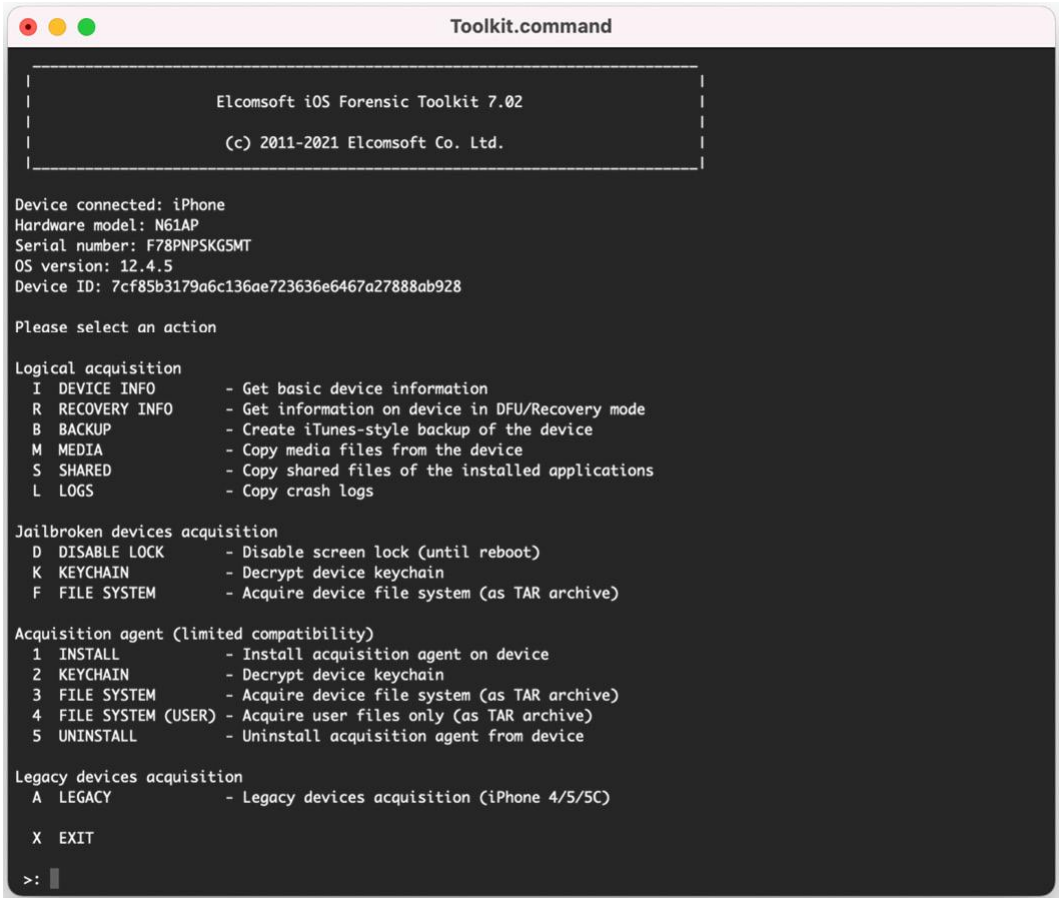


Figure 2.22 – Elcomsoft iOS Forensic Toolkit home screen

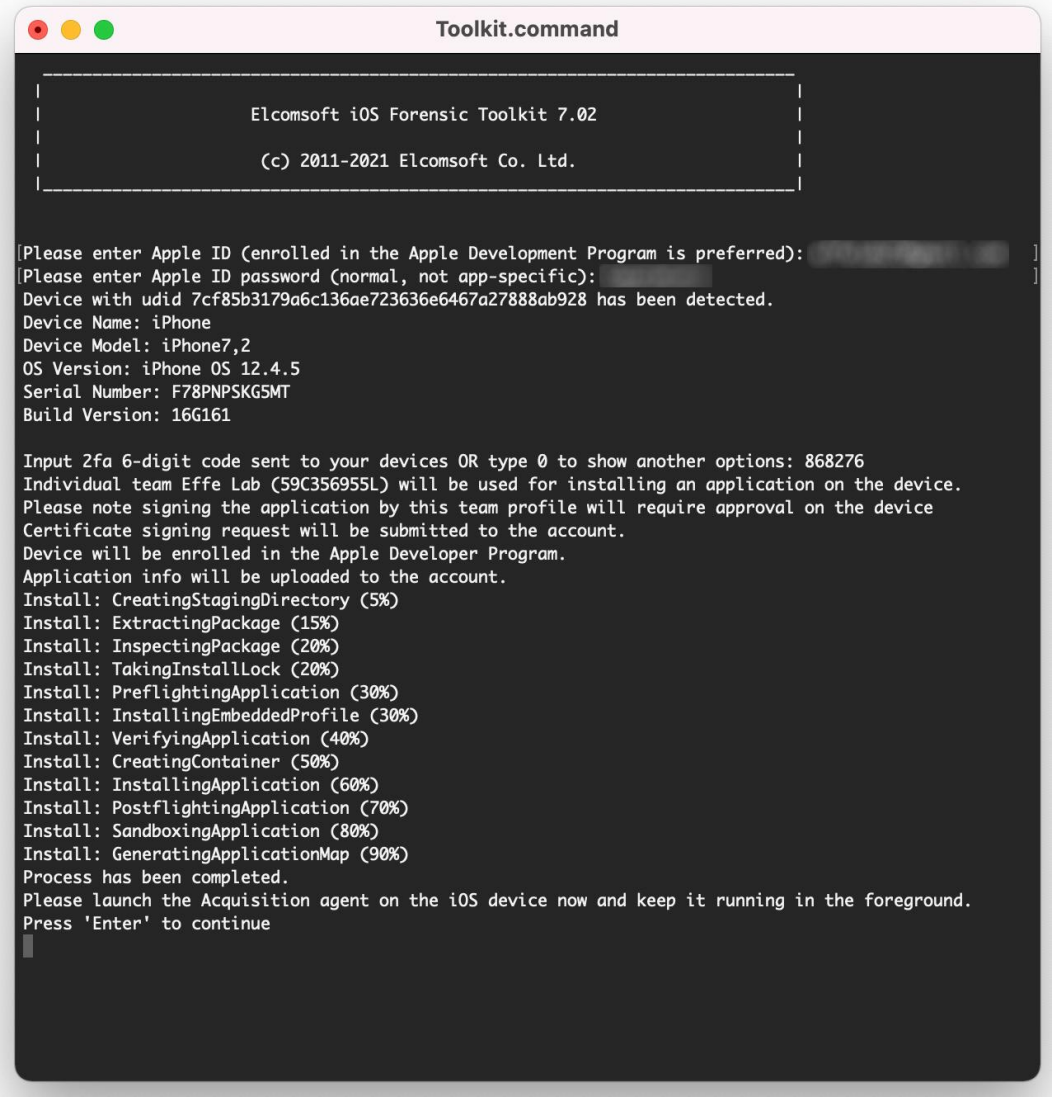


Figure 2.23 – Installing the agent onto the device

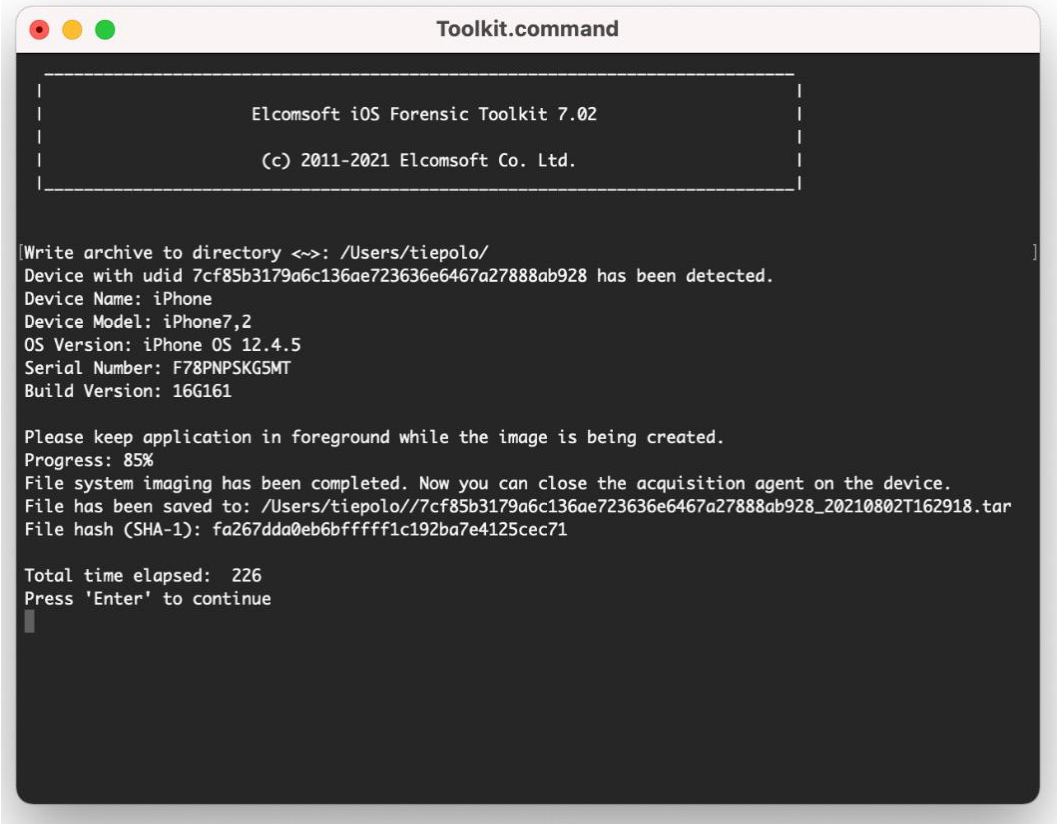


Figure 2.24 – Acquiring the full filesystem using the agent

Tables

	checkra1n	unc0ver
iPhone 6/6 Plus/6s/ 6s Plus	iOS 8–12.5 ✓	iOS 11–12.5 ✓
iPhone SE	iOS 9–14.8 ✓	iOS 11–14.3 ✓
iPhone 7/7 Plus	iOS 10–14.8 ✓	iOS 11–14.3 ✓
iPhone 8/8 Plus	iOS 11–14.8 ✓	iOS 11–14.3 ✓
iPhone X	iOS 11–14.8 ✓	iOS 11–14.3 ✓
iPhone XR	×	iOS 12–14.3 ✓
iPhone XS/XS Max	×	iOS 12–14.3 ✓
iPhone 11/11 Pro/11 Pro Max	×	iOS 13–14.3 ✓
iPhone 12/12 Pro/12 Pro Max	×	iOS 14–14.3 ✓
iPhone 13	×	×

Table 2.1 – iPhone model compatibility with checkra1n and unc0ver

Code and Commands

Command 2.1:

```
brew install libimobiledevice
```

Command 2.2:

```
sudo apt-get install usbmuxd libimobiledevice6
libimobiledevice-utils
```

Command 2.3

```
ideviceinfo
```

Command 2.4

```
ideviceinfo | grep ModelNumber
```

Command 2.5:

```
ideviceinfo | grep ProductVersion
```

Command 2.6:

```
ideviceinfo | grep SerialNumber
```

Command 2.7:

```
ideviceinfo | grep PhoneNumber
```

Chapter 3

Images

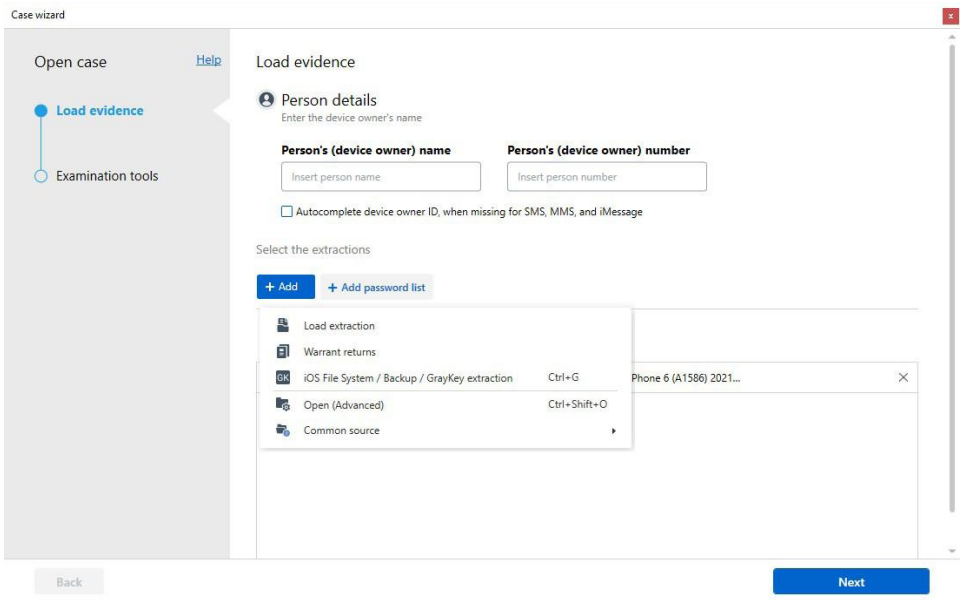


Figure 3.1 – Open the case wizard and add the folder or file that contains your data

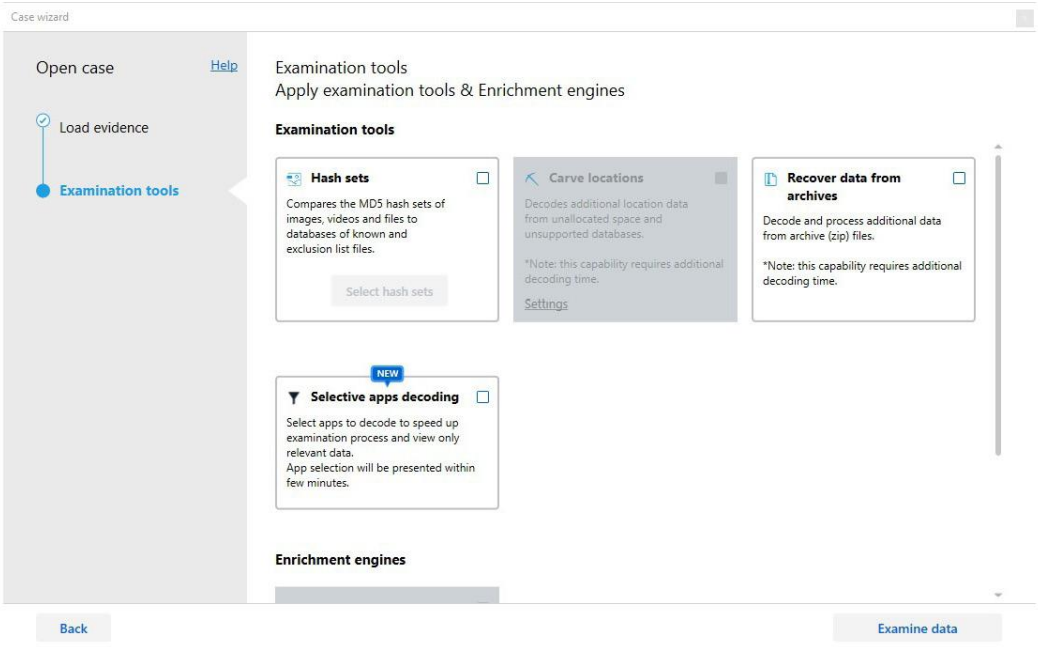


Figure 3.2 – The case wizard also allows you to enable specific examination tools

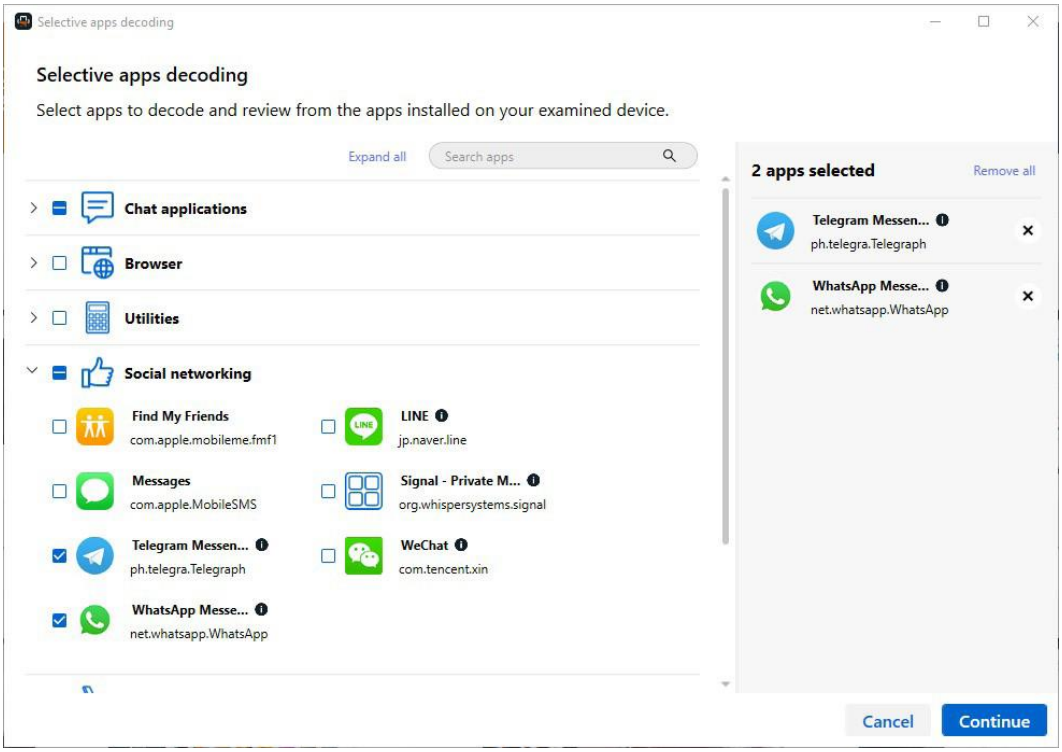


Figure 3.3 – Choose which applications should be decoded

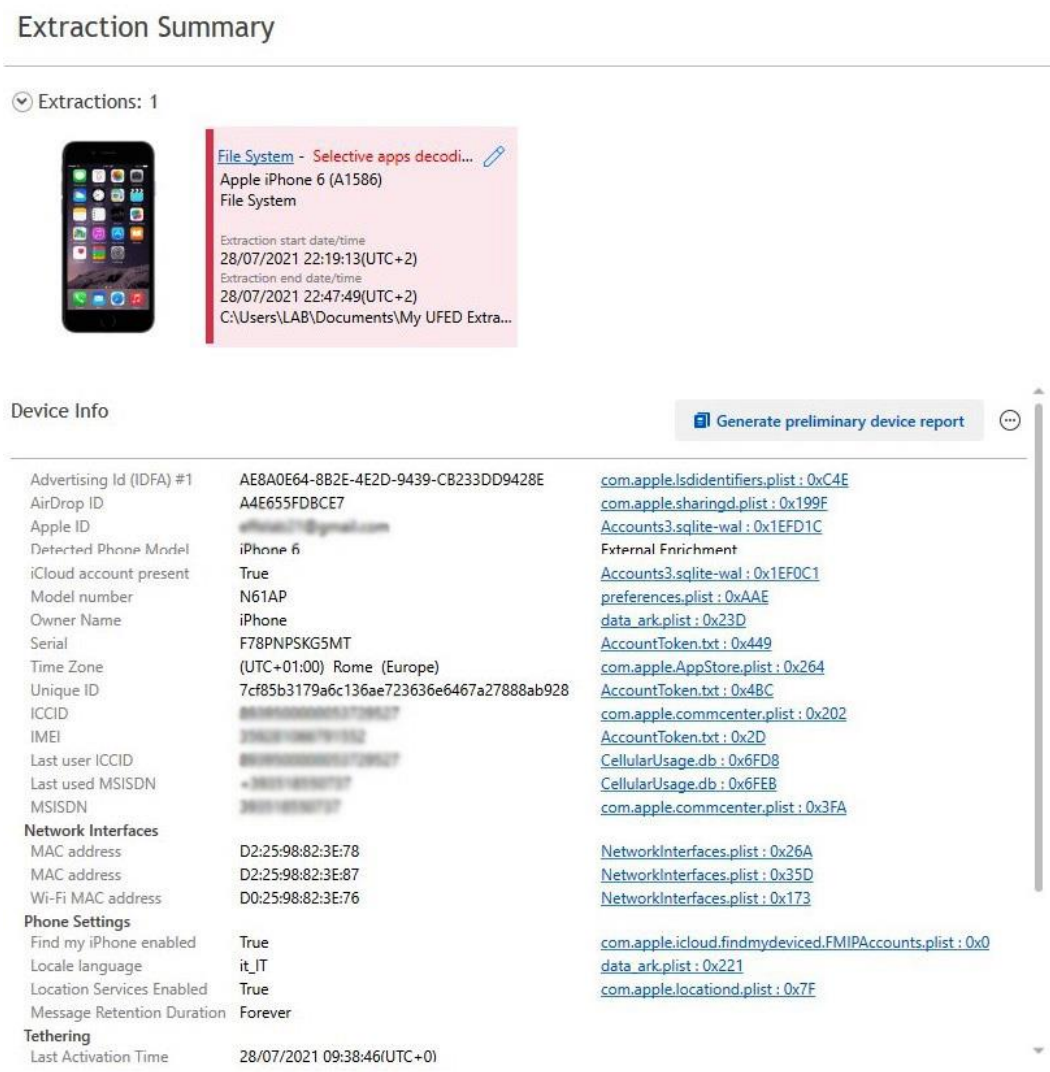


Figure 3.4 – Extraction Summary

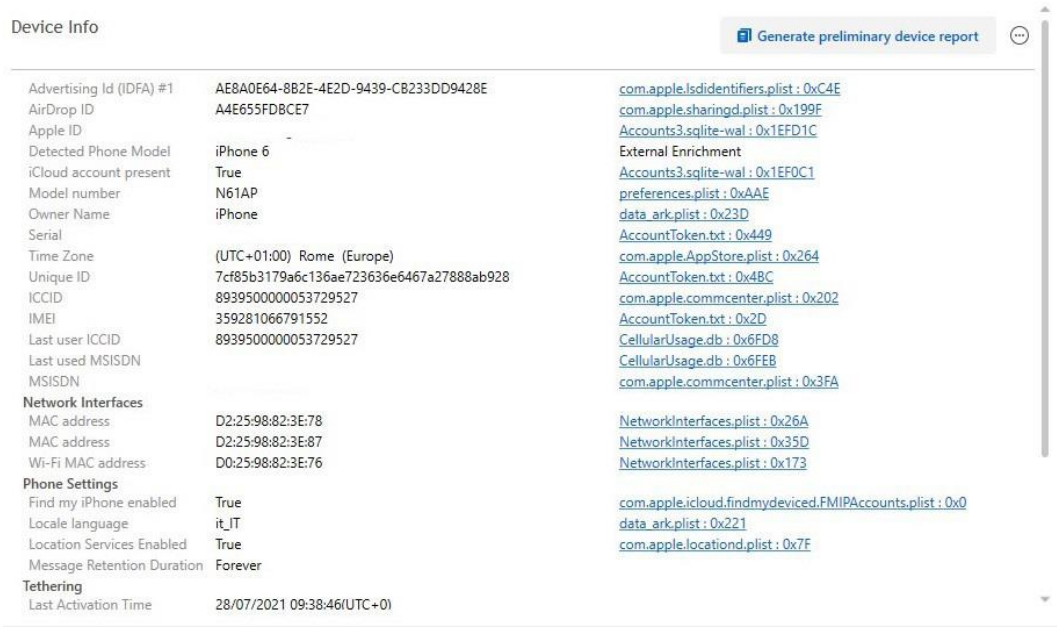


Figure 3.5 – The Device Info section gives the investigator some key data on the device

Figure 3.6 – The project tree displays all decoded data grouped into categories

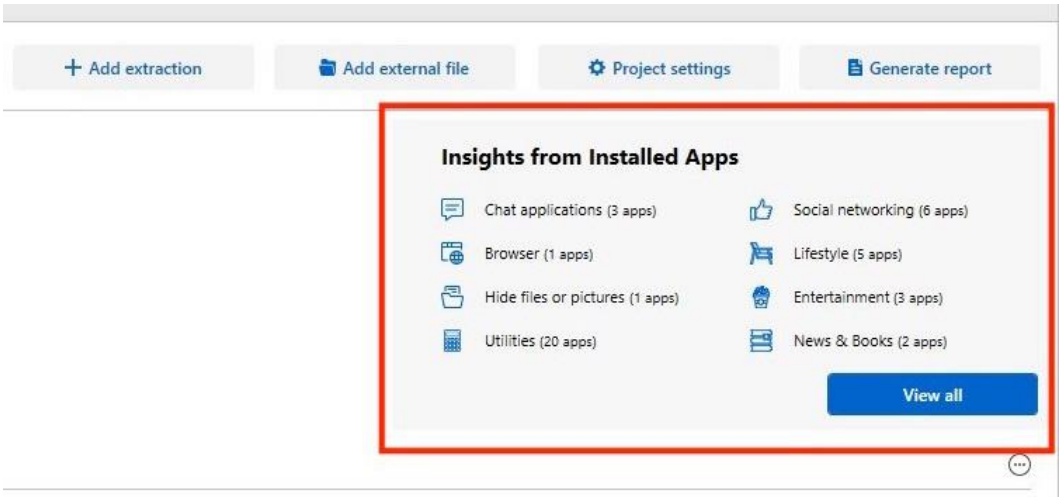


Figure 3.7 – Physical Analyzer extraction summary

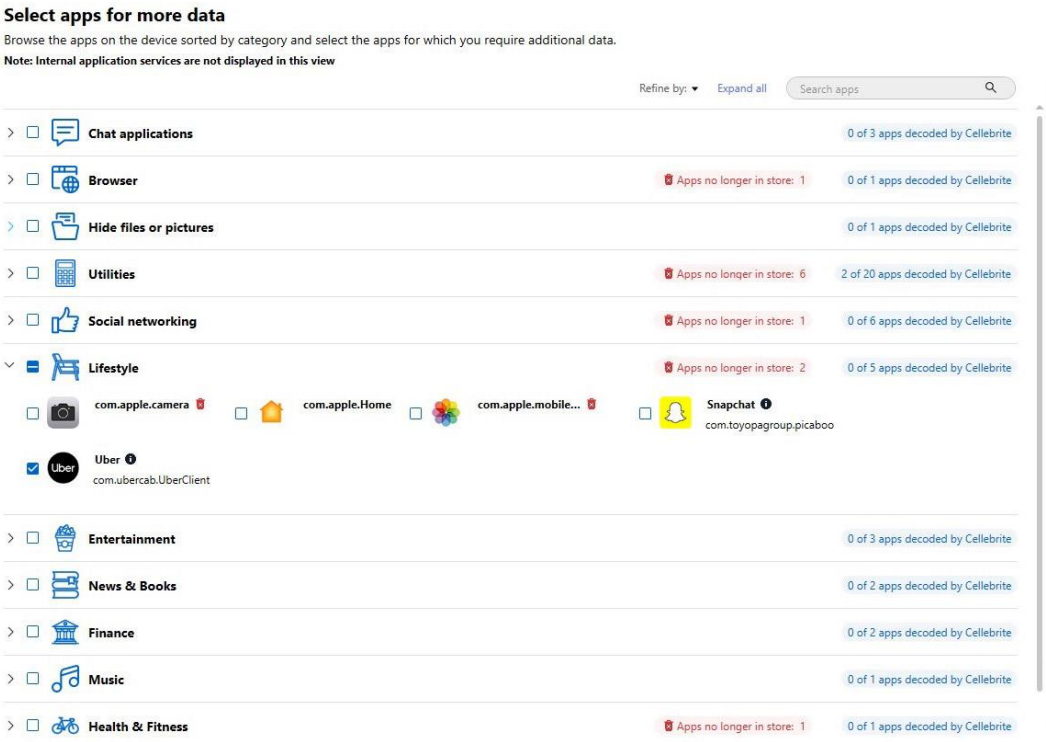


Figure 3.8 – A list of applications installed on the device

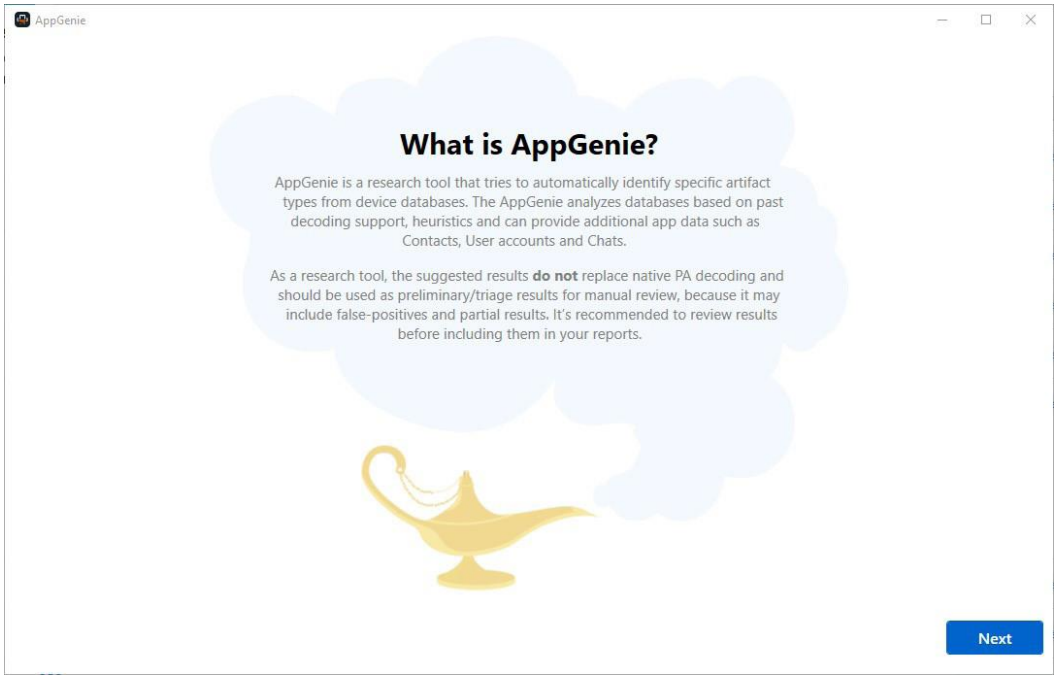


Figure 3.9 – A reminder that validation is required

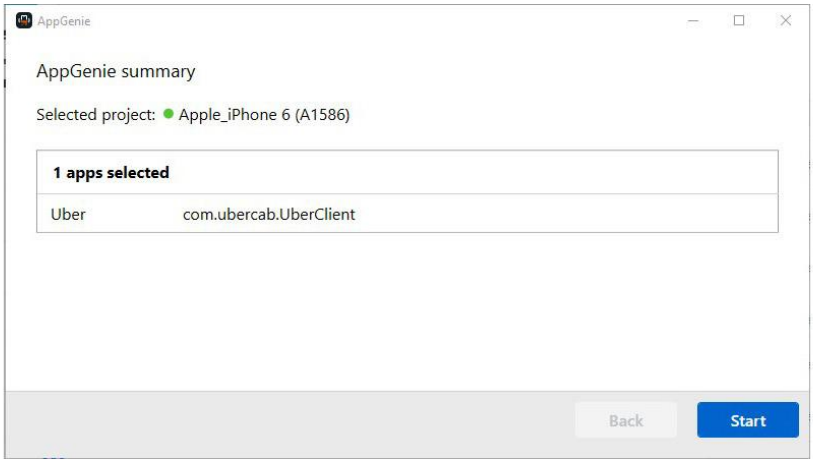


Figure 3.10 – A summary of applications that should be parsed by the AppGenie

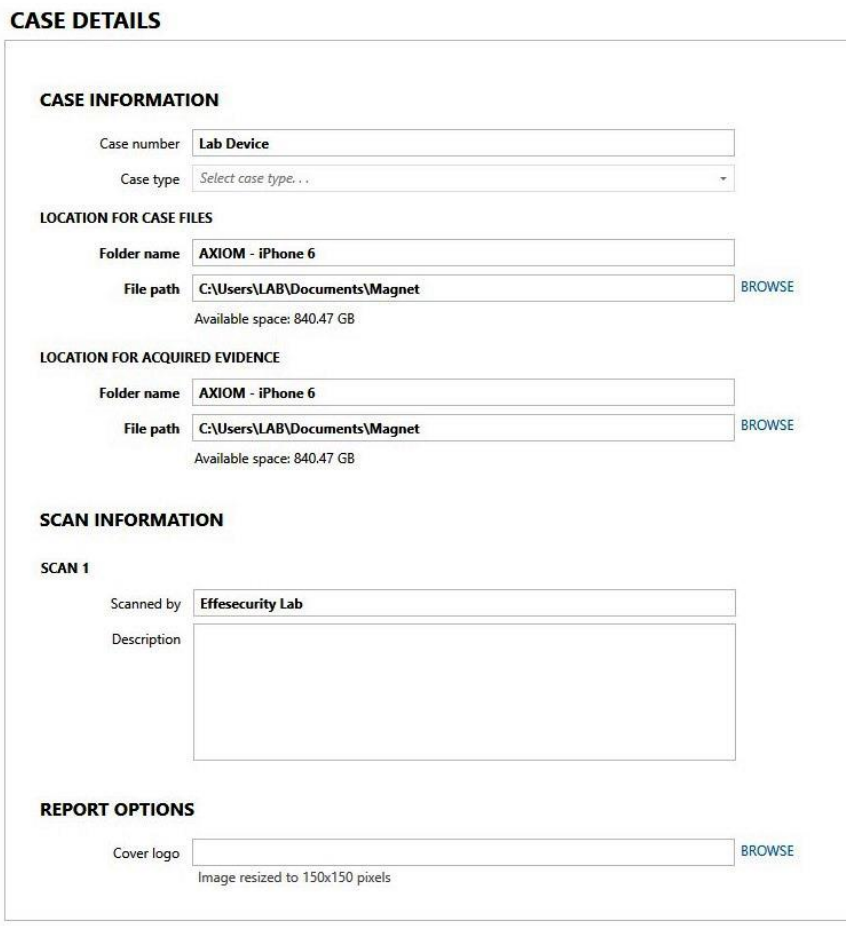


Figure 3.11 – Fill in the case details

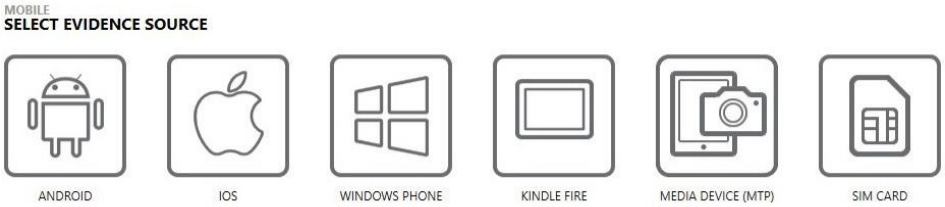


Figure 3.12 – Select the evidence source

Figure 3.13 – Choose the file or folder that contains the extraction

Figure 3.14 – The EVIDENCE SOURCES screen displays all data sources for the case

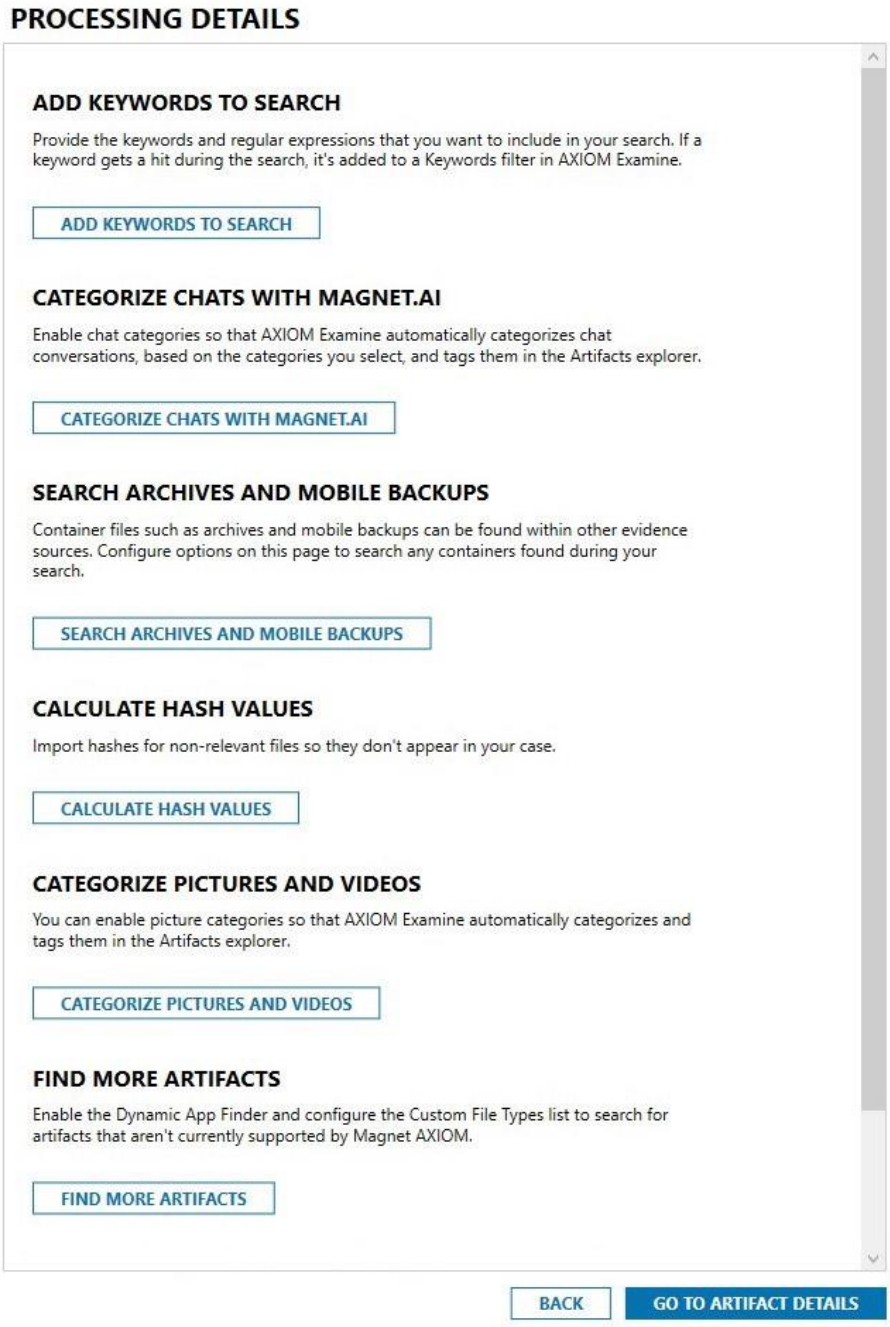


Figure 3.15 – Enable additional parsing tools and utilities

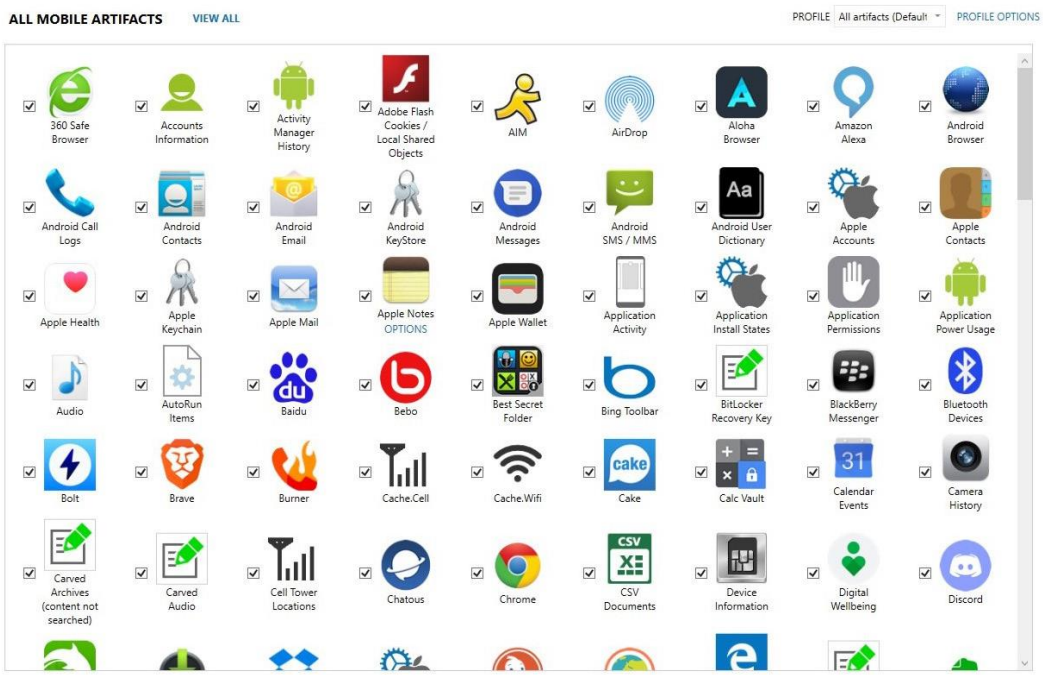


Figure 3.16 – Choose which artifacts should be decoded

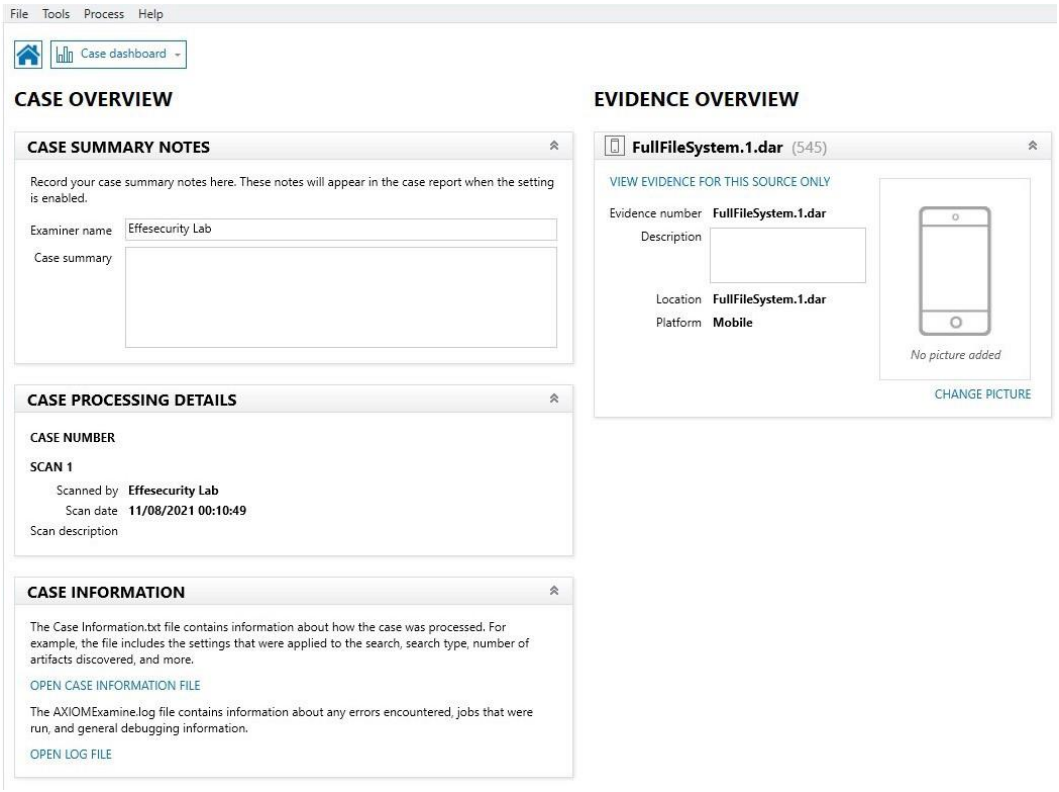


Figure 3.17 – The Case dashboard screen

EVIDENCE (26)

Column view

Potential Activity	Artifact	Artif...	Source
At Facebook home page	WebKit Browser Web History (Carved)	108	FullFileSystem.1.dar\Applications\MobileSafari.app\BuiltInBookmarkItems.plist
At Facebook home page	Safari History	149	FullFileSystem.1.dar\Applications\MobileSafari.app\BuiltInBookmarkItems.plist
At Facebook home page	Safari History	149	FullFileSystem.1.dar\Applications\MobileSafari.app\BuiltInBookmarkItems.plist
Unknown	Potential Browser Activity	11617	FullFileSystem.1.dar\private\var\db\uuidtext\9E\E12E6B290D344DA2002AECB9...
Looking at Facebook legal information	Potential Browser Activity	11623	FullFileSystem.1.dar\private\var\db\uuidtext\9E\E12E6B290D344DA2002AECB9...
Looking at Facebook legal information	Potential Browser Activity	11621	FullFileSystem.1.dar\private\var\db\uuidtext\9E\E12E6B290D344DA2002AECB9...
Looking at Facebook legal information	Potential Browser Activity	11631	FullFileSystem.1.dar\private\var\db\uuidtext\49\D116B807F93F48AD657DE2768...
At Facebook home page	KnowledgeC Application Web Usage	14397	FullFileSystem.1.dar\private\var\mobile\Library\CoreDuet\Knowledge\knowledg...
At Facebook login page	KnowledgeC Application Web Usage	14397	FullFileSystem.1.dar\private\var\mobile\Library\CoreDuet\Knowledge\knowledg...
At Facebook home page	KnowledgeC Application Web Usage	14404	FullFileSystem.1.dar\private\var\mobile\Library\CoreDuet\Knowledge\knowledg...
At Facebook login page	KnowledgeC Application Web Usage	14404	FullFileSystem.1.dar\private\var\mobile\Library\CoreDuet\Knowledge\knowledg...
At Facebook home page	KnowledgeC Application Web Usage	14407	FullFileSystem.1.dar\private\var\mobile\Library\CoreDuet\Knowledge\knowledg...
At Facebook login page	KnowledgeC Application Web Usage	14407	FullFileSystem.1.dar\private\var\mobile\Library\CoreDuet\Knowledge\knowledg...
At Facebook home page	KnowledgeC Application Web Usage	14410	FullFileSystem.1.dar\private\var\mobile\Library\CoreDuet\Knowledge\knowledg...
Unknown	KnowledgeC Application Web Usage	14410	FullFileSystem.1.dar\private\var\mobile\Library\CoreDuet\Knowledge\knowledg...
Unknown	KnowledgeC Safari History	15102	FullFileSystem.1.dar\private\var\mobile\Library\CoreDuet\Knowledge\knowledg...
At Facebook login page	KnowledgeC Safari History	15119	FullFileSystem.1.dar\private\var\mobile\Library\CoreDuet\Knowledge\knowledg...
At Facebook login page	KnowledgeC Safari History	15161	FullFileSystem.1.dar\private\var\mobile\Library\CoreDuet\Knowledge\knowledg...
Unknown	Safari History	22789	FullFileSystem.1.dar\private\var\mobile\Containers\Data\Application\45D7E9EB...
At Facebook login page	Safari History	22789	FullFileSystem.1.dar\private\var\mobile\Containers\Data\Application\45D7E9EB...
At Facebook login page	Safari History	22791	FullFileSystem.1.dar\private\var\mobile\Containers\Data\Application\45D7E9EB...
Unknown	Safari History	22950	FullFileSystem.1.dar\private\var\mobile\Containers\Data\Application\45D7E9EB...
At Facebook login page	Safari History	22952	FullFileSystem.1.dar\private\var\mobile\Containers\Data\Application\45D7E9EB...
Unknown	Potential Browser Activity	23110	FullFileSystem.1.dar\private\var\mobile\Containers\Data\Application\45D7E9EB...
Unknown	Safari History	23191	FullFileSystem.1.dar\private\var\mobile\Containers\Data\Application\45D7E9EB...
At Facebook login page	Safari History	23193	FullFileSystem.1.dar\private\var\mobile\Containers\Data\Application\45D7E9EB...

Figure 3.18 – The Artifacts explorer

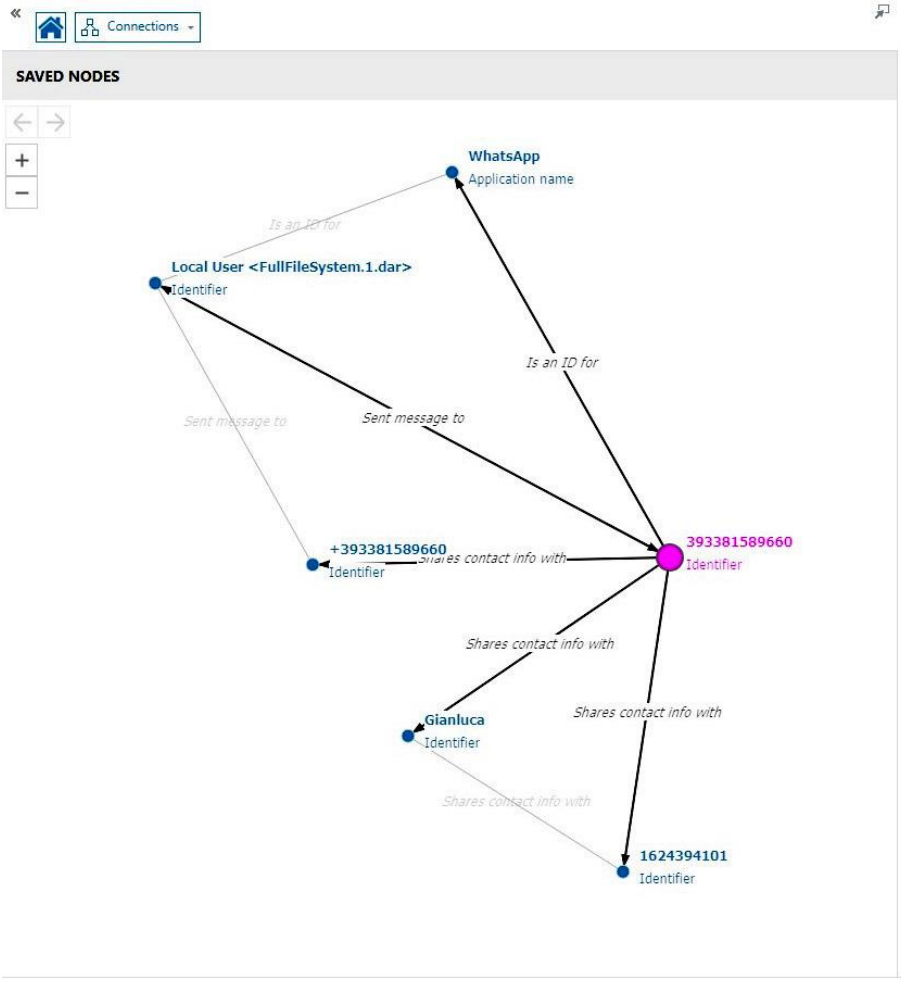


Figure 3.19 – The Connections explorer

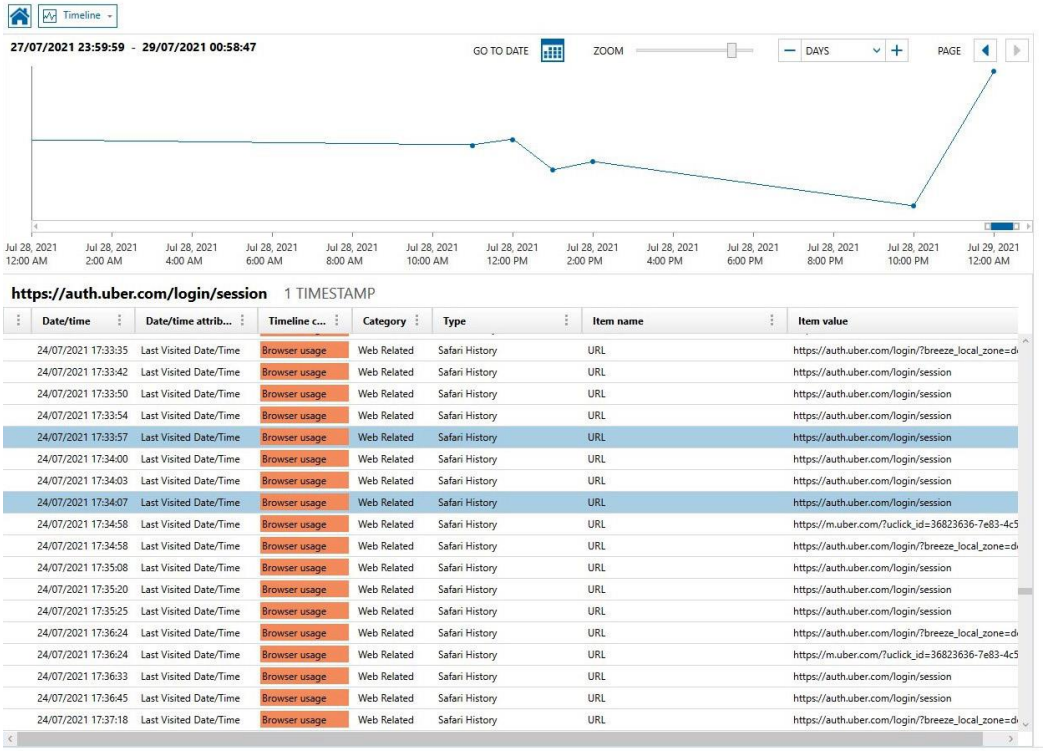


Figure 3.20 – The Timeline explorer

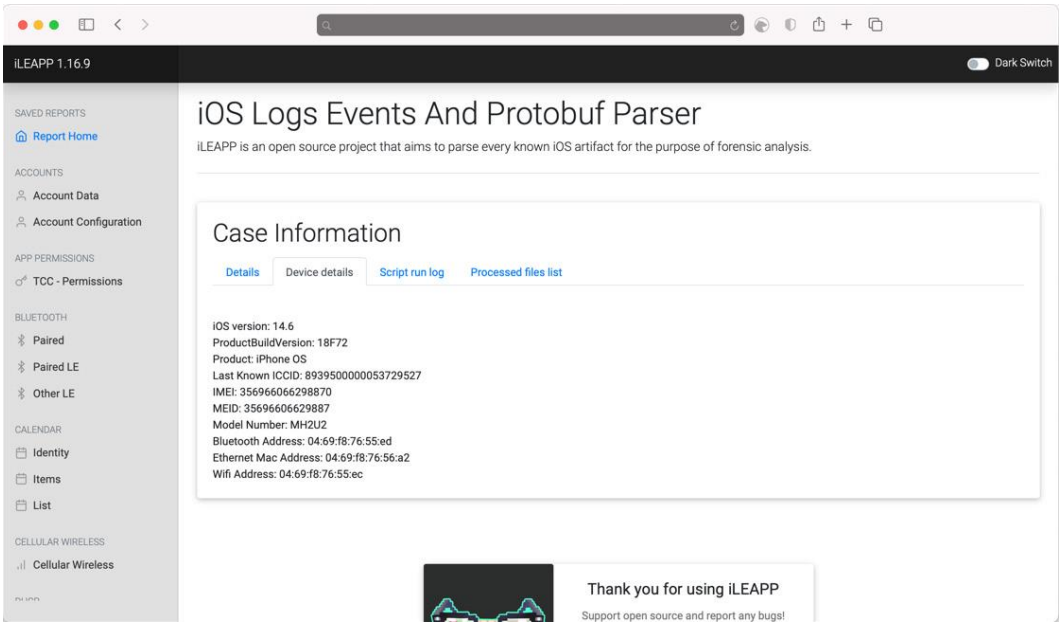


Figure 3.21 – The report generated by iLEAPP

Code and Commands

Command 3.1

```
git clone https://github.com/abrignoni/iLEAPP.git
```

Command 3.2:

```
cd iLEAPP

pip install -r requirements.txt
```

Command 3.3:

```
python3 ileapp.py -t zip -i ../iphone_dump.zip -o output
```

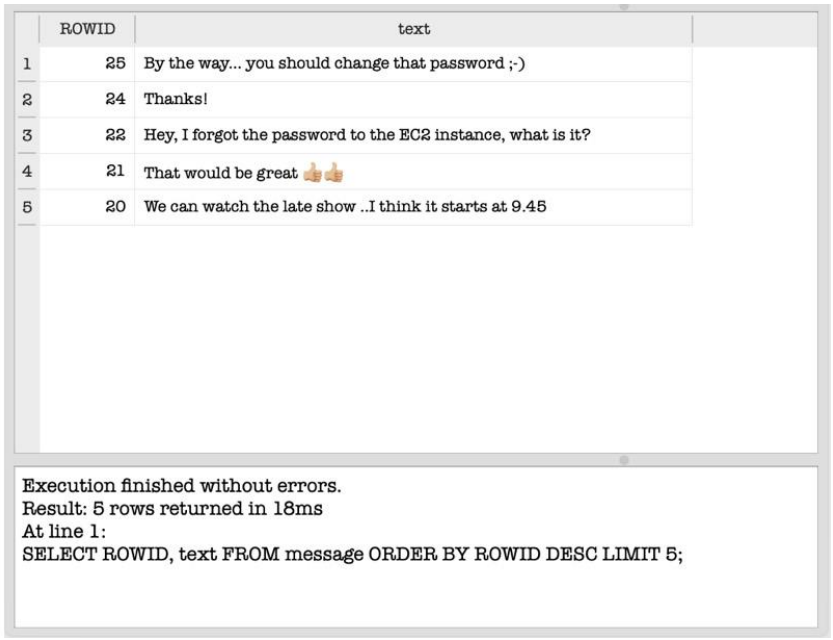
Links

- You can download APOLLO from the project's GitHub repository: https://github.com/mac4n6/APOLLO

- iLEAPP can be downloaded from https://github.com/abrignoni/iLEAPP.
- You can download this iOS Triage from GitHub: https://github.com/RealityNet/ios_triage.
- You can download the Sysdiagnose scripts from https://github.com/cheeky4n6monkey/iOS_sysdiagnose_forensic_scripts.

Chapter 4

Images



	ROWID	text
1	25	By the way... you should change that password ;-)
2	24	Thanks!
3	22	Hey, I forgot the password to the EC2 instance, what is it?
4	21	That would be great 👍👍
5	20	We can watch the late show ..I think it starts at 9.45

Execution finished without errors.
Result: 5 rows returned in 18ms
At line 1:
SELECT ROWID, text FROM message ORDER BY ROWID DESC LIMIT 5;

Figure 4.1 – The result of a query displayed in DB Browser for SQLite

```
00000000  53 51 4c 69 74 65 20 66  6f 72 6d 61 74 20 33 00  |SQLite format 3.|
00000010  10 00 02 02 00 40 20 20  00 00 00 13 00 00 00 43  |.....@  .......C|
00000020  00 00 00 00 00 00 00 00  00 00 00 50 00 00 00 04  |...........P....|
00000030  00 00 00 00 00 00 00 00  00 00 00 01 00 00 00 00  |................|
00000040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 13  |................|
00000060  00 2e 24 80                                       |..$.|
00000064
```

Figure 4.2 – The output from the hexdump command

Offset	Size	Description
0	16	The header string: "SQLite format 3\000"
16	2	The database page size in bytes.
18	1	File format write version. 1 for legacy; 2 for WAL.
19	1	File format read version. 1 for legacy; 2 for WAL.
20	1	Bytes of unused "reserved" space at the end of each page. Usually 0.
21	1	Maximum embedded payload fraction. Must be 64.
22	1	Minimum embedded payload fraction. Must be 32.
23	1	Leaf payload fraction. Must be 32.
24	4	File change counter.
28	4	Size of the database file in pages. The "in-header database size".
32	4	Page number of the first freelist trunk page.
36	4	Total number of freelist pages.
40	4	The schema cookie.
44	4	The schema format number. Supported schema formats are 1, 2, 3, and 4.
48	4	Default page cache size.
52	4	The page number of the largest root b-tree page when in auto-vacuum or zero otherwise.
56	4	The database text encoding. A value of 1 means UTF-8. A value of 2 means UTF-16le. A value of 3 means UTF-16be.
60	4	The "user version" as read and set by the user_version pragma.
64	4	True (non-zero) for incremental-vacuum mode. False (zero) otherwise.
68	4	The "Application ID" set by PRAGMA application_id.
72	20	Reserved for expansion. Must be zero.
92	4	The version-valid-for number.
96	4	SQLITE_VERSION_NUMBER

Figure 4.3 – The SQLite database header reference

Offset	Size	Description
0	1	The one-byte flag at offset 0 indicating the b-tree page type. • A value of 2 (0x02) means the page is an interior index b-tree page. • A value of 5 (0x05) means the page is an interior table b-tree page. • A value of 10 (0x0a) means the page is a leaf index b-tree page. • A value of 13 (0x0d) means the page is a leaf table b-tree page.
1	2	The two-byte integer at offset 1 gives the start of the first freeblock on the page, or is zero if there are no freeblocks.
3	2	The two-byte integer at offset 3 gives the number of cells on the page.
5	2	The two-byte integer at offset 5 designates the start of the cell content area. A zero value for this integer is interpreted as 65536.
7	1	The one-byte integer at offset 7 gives the number of fragmented free bytes within the cell content area.
8	4	The four-byte page number at offset 8 is the right-most pointer. This value appears in the header of interior b-tree pages only and is omitted from all other pages.

Figure 4.4 – The SQLite page header reference

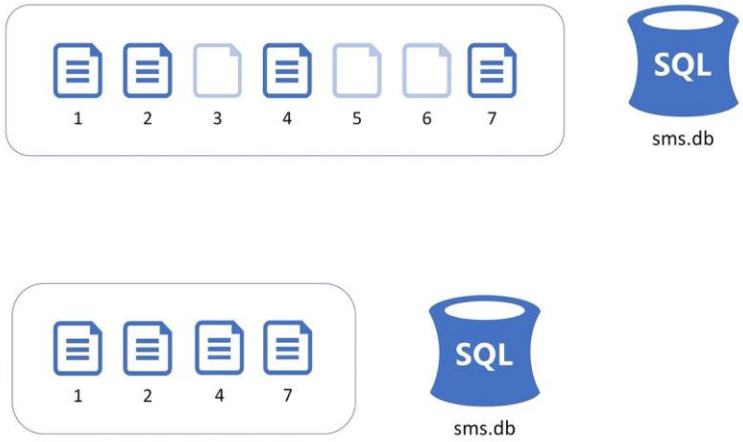


Figure 4.5 – Pages in a fragmented SQLite database

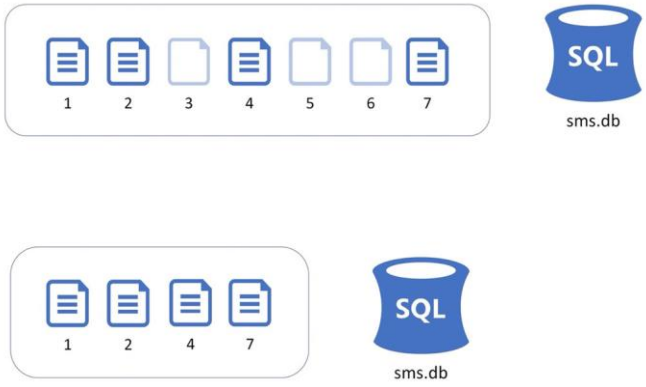


Figure 4.6 – Pages in an SQLite database after the vacuuming process

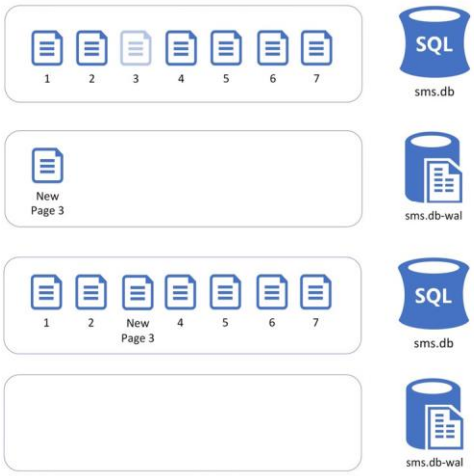


Figure 4.7 – New records added to the database are appended to the WAL file

Figure 4.8 – A checkpoint transfers data from the WAL file back into the main database

Offset	Size	Description
0	4	Magic number. 0x377f0682 or 0x377f0683
4	4	File format version. Currently 3007000.
8	4	Database page size. Example: 1024
12	4	Checkpoint sequence number
16	4	Salt-1: random integer incremented with each checkpoint
20	4	Salt-2: a different random number for each checkpoint
24	4	Checksum-1: First part of a checksum on the first 24 bytes of header
28	4	Checksum-2: Second part of the checksum on the first 24 bytes of header

Figure 4.9 – The WAL file header

Offset	Size	Description
0	4	Page number
4	4	The size of the database file in pages after the commit
8	4	Salt-1 copied from the WAL header
12	4	Salt-2 copied from the WAL header
16	4	Checksum-1: Cumulative checksum up through and including this page
20	4	Checksum-2: Second half of the cumulative checksum.

Figure 4.10 – The WAL frame header

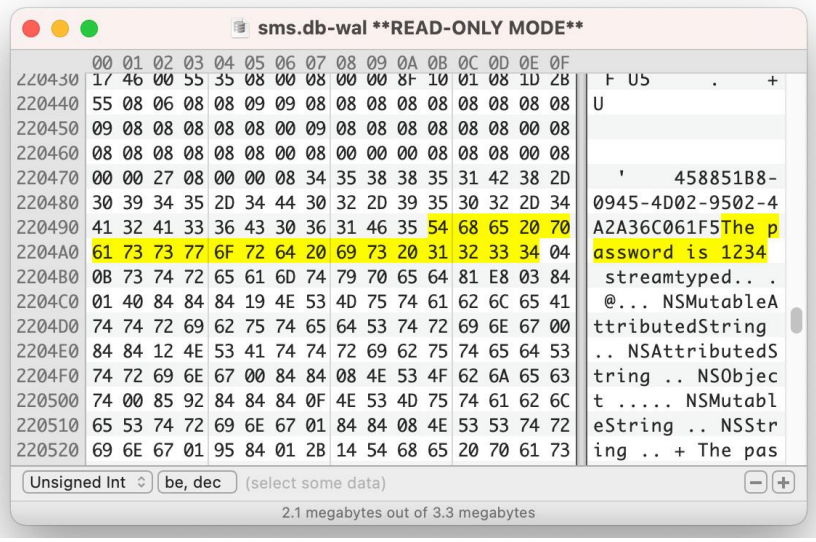


Figure 4.11 – Using a hex viewer to display the contents of an SQLite file

```
1  SELECT * FROM Fruits;
```

	ROWID	Fruit
1	2	Peach
2	3	Orange
3	4	Banana

Figure 4.12 – The table contains three rows of data

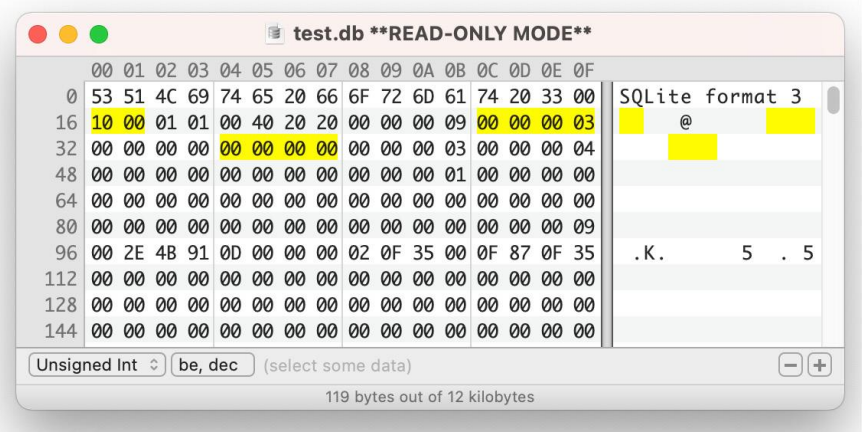


Figure 4.13 – The SQLite database header

Page	Offset (Dec)	Offset (HEX)
1	0	0x0000
2	4096	0x1000
3	8192	0x2000

Figure 4.14 – The decimal offset for each page

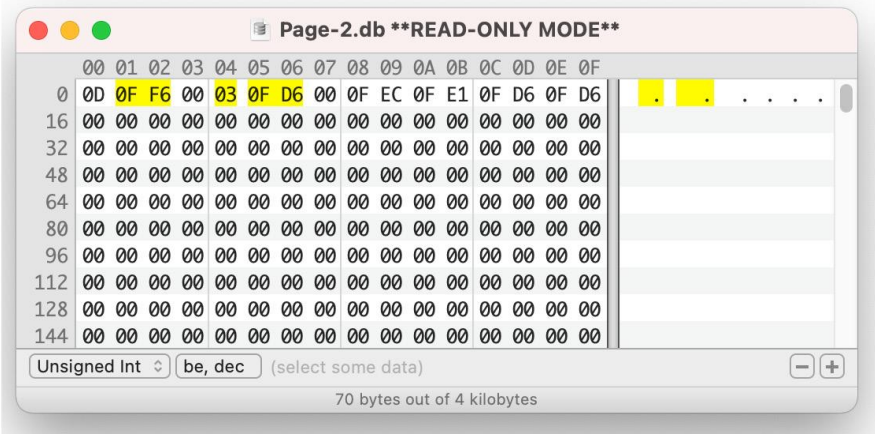


Figure 4.15 – The page header

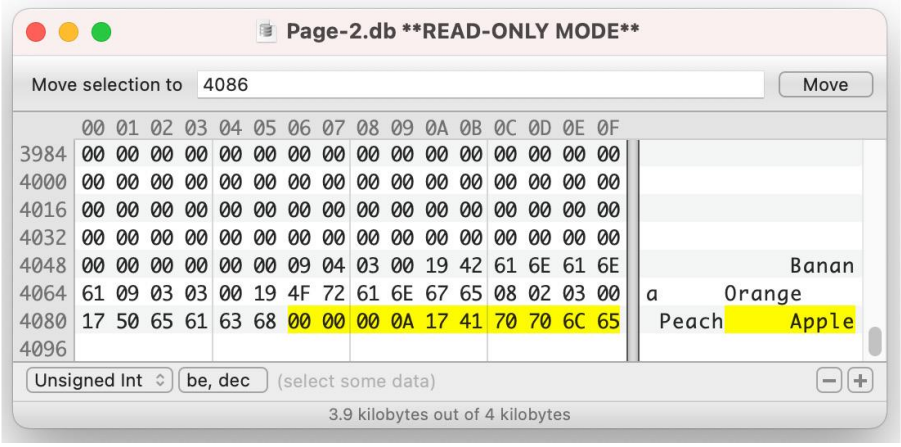


Figure 4.16 – The free block shows previously deleted data

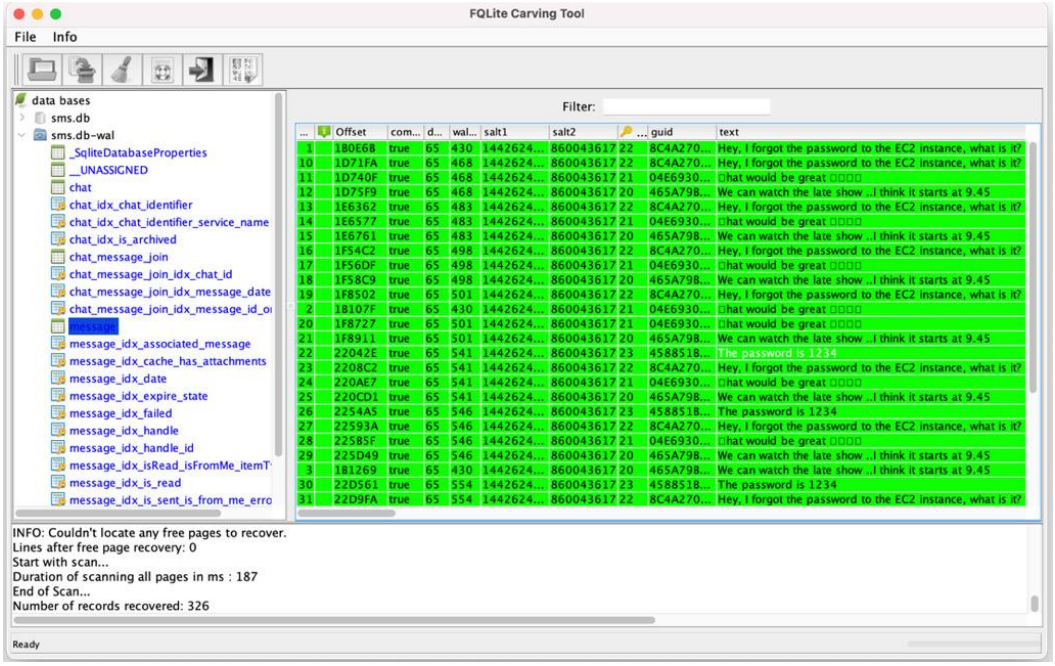


Figure 4.17 – FQLite recovers deleted records from an SQLite database

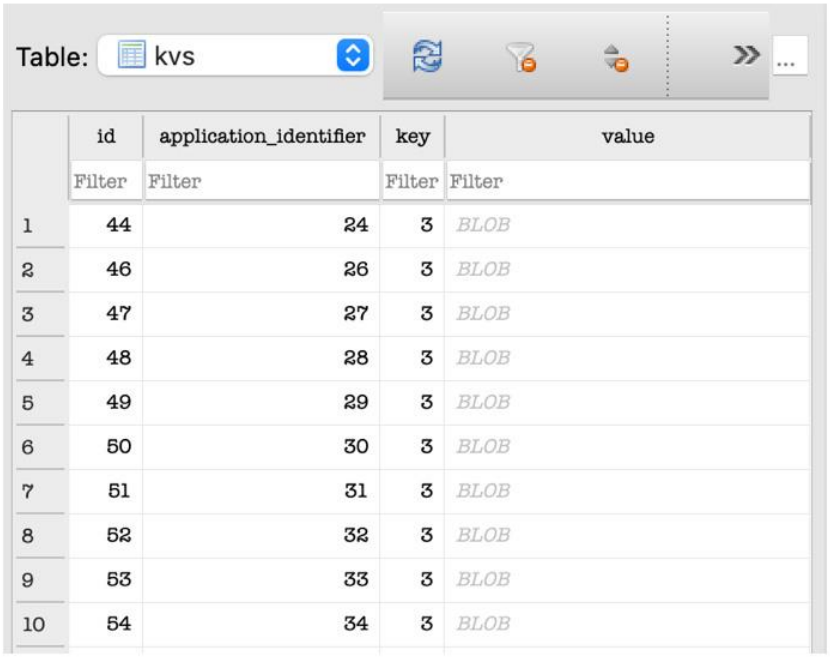


Figure 4.18 – A SQLite database containing BLOBs of binary data

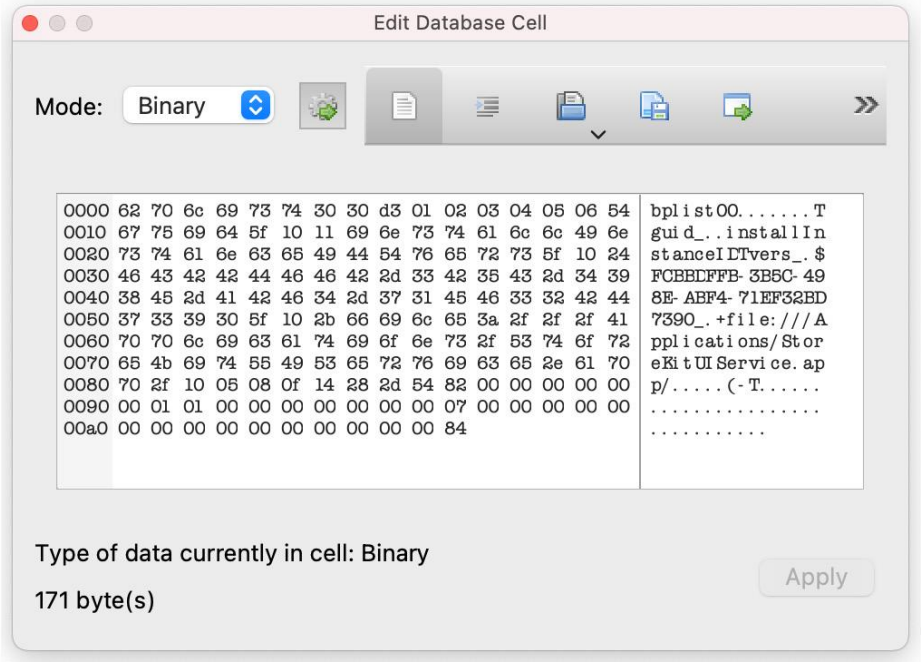


Figure 4.19 – The hex viewer shows the bplist in binary format

Path	Description
installd/Library/MobileInstallation/LastBuildInfo.plist	OS version
mobile/Library/Preferences/com.apple.springboard.plist	Order of apps
mobile/Library/Preferences/com.apple.mobilegestalt.plist	Device name
mobile/Library/Preferences/com.apple.Preferences.plist	Disk usage
mobile/Library/Preferences/com.apple.purplebuddy.plist	Language
mobile/Library/Preferences/com.apple.commcenter.shared.plist	Phone number
mobile/Library/Preferences/com.apple.corerecents.recentsd.plist	iCloud data
preferences/SystemConfiguration/com.apple.networkidentification.plist	Network data
preferences/SystemConfiguration/NetworkInterfaces.plist	WiFi data
preferences/SystemConfiguration/preferences.plist	Cellular + WiFi

Figure 4.20 – Common iOS device artifacts

Path	Description
mobile/Library/Mail/Recents.db	Recent SMS and email metadata
mobile/Media/Recordings/Recordings.db	Audio recordings
mobile/Media/PhotoData/Photos.sqlite	Photo categorization
mobile/Library/CallHistoryDB/CallHistory.storedata	Call log
mobile/Library/SMS/sms.db	SMS/MMS/iMessage
mobile/Library/Safari/History.db	Browsing history
mobile/Media/DCIM/*	Photos
mobile/Library/Accounts/Accounts3.sqlite	Account details

Figure 4.21 – Common iOS user artifacts

Code and Commands

Code 4.1:

```
SELECT ROWID, text FROM message

ORDER BY ROWID DESC

LIMIT 5;
```

Command 4.1:

```
sqlite3 filename.db
```

Command 4.2:

```
sqlite> .tables

attachment                  message
```

```
chat                        message_attachment_join
chat_handle_join            message_processing_task
chat_message_join           sync_deleted_attachments
deleted_messages            sync_deleted_chats
```

Command 4.3:

```
sqlite> .schema handle
CREATE TABLE handle (ROWID INTEGER PRIMARY KEY
AUTOINCREMENT UNIQUE, id TEXT NOT NULL, country TEXT,
service TEXT NOT NULL);
```

Command 4.4:

```
SELECT * FROM handle;
```

Command 4.5:

```
hexdump -n 100 -C sms.db
```

Command 4.6:

```
CREATE TABLE "Fruits" (
     "ROWID"     INTEGER,
     "Fruit"     TEXT,
     PRIMARY KEY("ROWID" AUTOINCREMENT)
);
INSERT INTO "Fruits" VALUES (1,'Apple');
INSERT INTO "Fruits" VALUES (2,'Peach');
INSERT INTO "Fruits" VALUES (3,'Orange');
INSERT INTO "Fruits" VALUES (4,'Banana');
DELETE FROM Fruits WHERE ROWID == 1;
```

Code 4.2:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
     <key>DCSAssetPreferenceKeyDownloadedDictionaries</key>
     <array>
           <string>Apple Dictionary.dictionary</string>
           <string>Italian.dictionary</string>
           <string>Italian - English.dictionary</string>
     </array>
</dict>
</plist>
```

Command 4.7:

```
plutil -convert xml1 filename.plist
```

Command 4.8:

```
protoc --decode_raw < [protobuf_blob_file]
```

Links

- To download FQLite, visit https://www.staff.hs-mittweida.de/~pawlaszc/fqlite/ or head over to the GitHub repository that is located at https://github.com/pawlaszczyk/fqlite.
- The `protoc` tool is part of the protocol buffers package and can be downloaded from the GitHub repository located at http://github.com/protocolbuffers/protobuf.

Chapter 5

Images

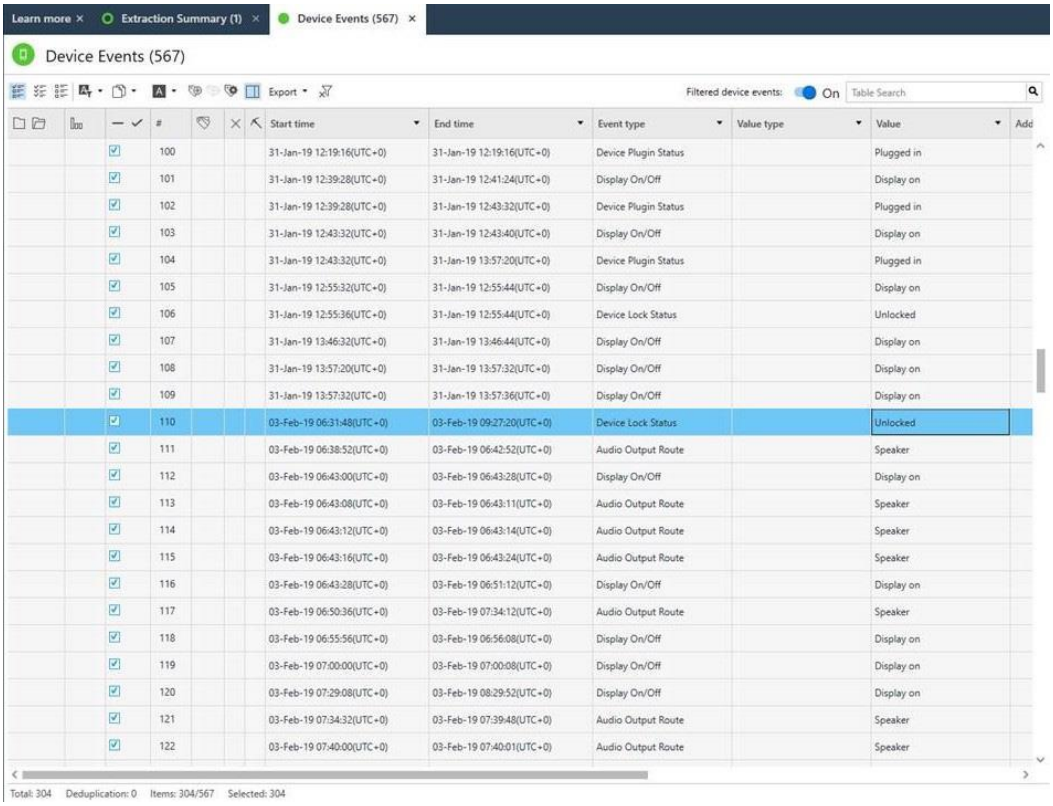


Figure 5.1 – Device events in Cellebrite Physical Analyzer

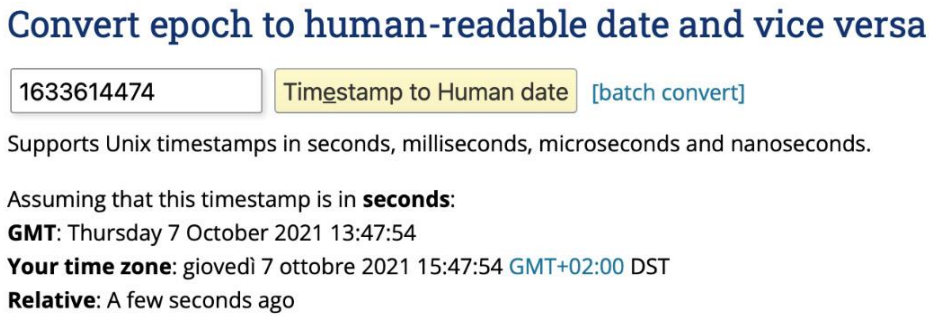


Figure 5.2 – Converting a Unix timestamp to a string

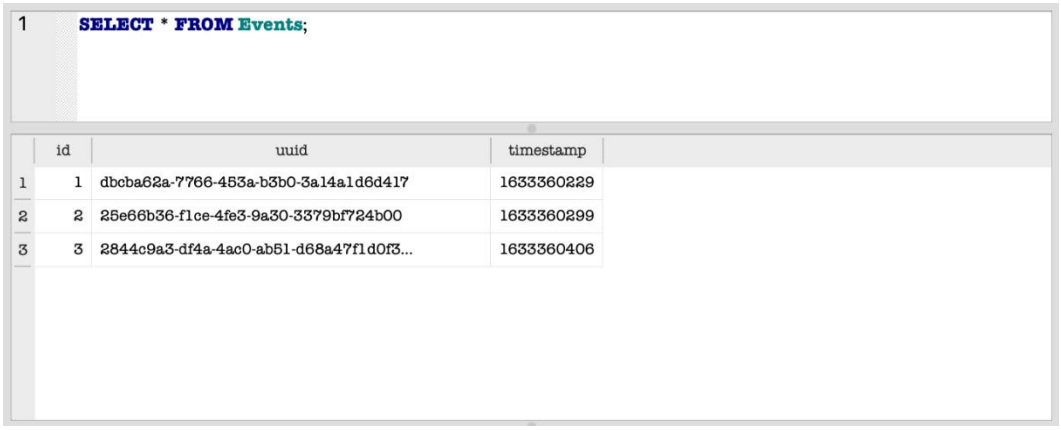


Figure 5.3 – The Events table in a SQLite database

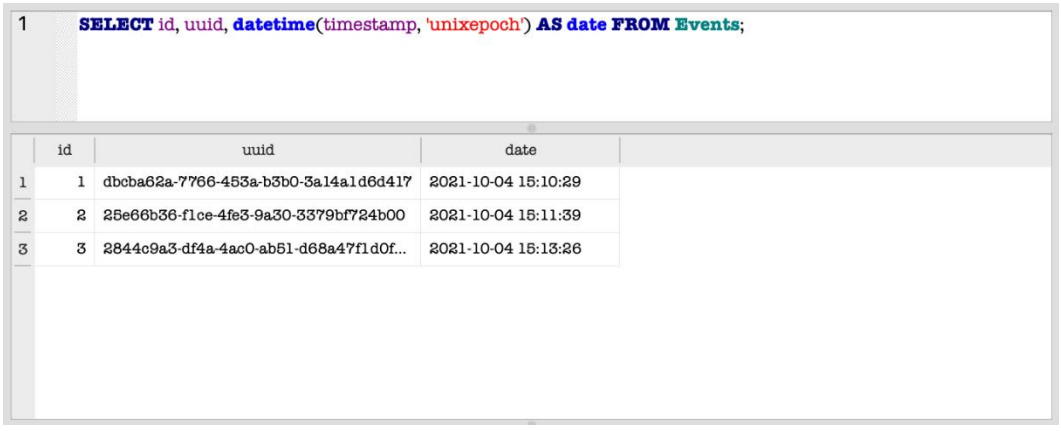


Figure 5.4 – Converting a timestamp to a string

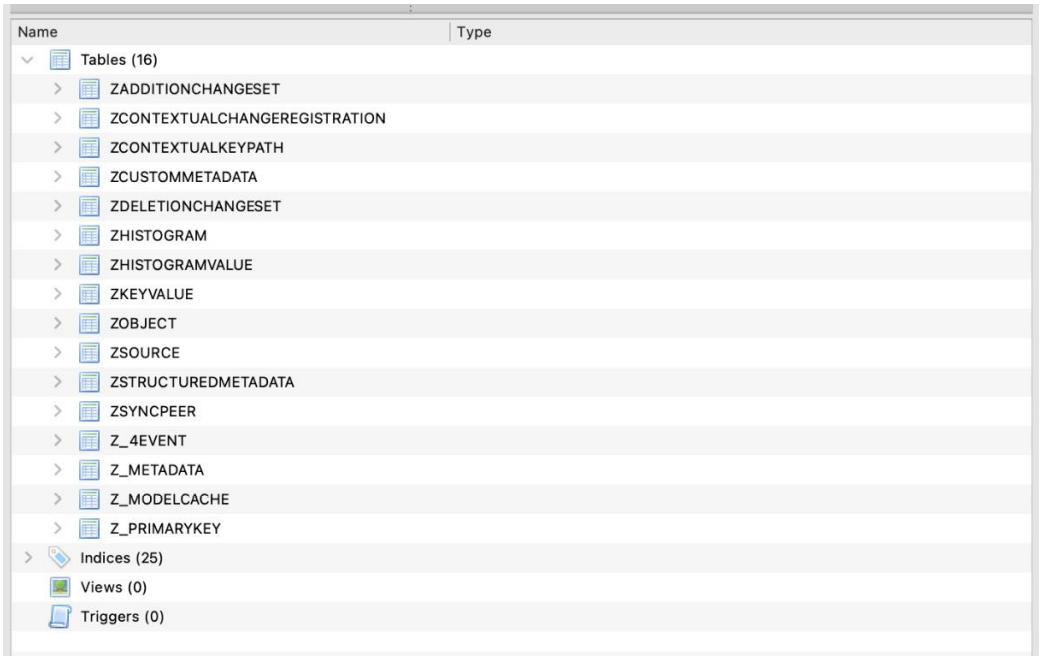


Figure 5.5 – The schema of the KnowledgeC database

Result

Conversion of date and time: **Monday, 2021-05-24 00:00:00 UTC**

Date and Time:	**Monday, 2021-05-24 00:00:00 UTC**
UNIX Epoch Time:	**1621814400**
UNIX Epoch Time (Hex):	**60AAEC80**
CF Absolute Time:	**643507200**

Figure 5.6 – Converting date and time to a MAC timestamp using https://www.gaijin.at/en/tools/time-converter

	START	END	GMT OFFSET	USAGE IN SECONDS	ZSTREAMNAME	ZVALUESTRING
1	2021-05-24 08:36:56	2021-05-24 08:37:10	2	14	/app/inFocus	com.apple.mobilesafari
2	2021-05-24 08:37:10	2021-05-24 08:37:12	2	2	/app/inFocus	com.apple.CoreAuthUI
3	2021-05-24 08:37:12	2021-05-24 08:37:32	2	20	/app/inFocus	com.apple.mobilesafari
4	2021-05-24 08:37:32	2021-05-24 08:37:34	2	2	/app/inFocus	us.zoom.videomeetings
5	2021-05-24 08:37:34	2021-05-24 08:37:37	2	3	/app/inFocus	com.apple.mobilesafari
6	2021-05-24 08:37:37	2021-05-24 08:38:13	2	36	/app/inFocus	us.zoom.videomeetings
7	2021-05-24 08:38:13	2021-05-24 10:56:11	2	8278	/app/inFocus	us.zoom.videomeetings
8	2021-05-24 10:56:15	2021-05-24 10:58:45	2	150	/app/inFocus	com.apple.mobilesafari
9	2021-05-24 10:56:17	2021-05-24 10:56:18	2	1	/app/inFocus	com.apple.PDFKit.PDFExtensionView
10	2021-05-24 10:56:19	2021-05-24 10:56:21	2	2	/app/inFocus	com.apple.PDFKit.PDFExtensionView
11	2021-05-24 10:58:45	2021-05-24 10:58:46	2	1	/app/inFocus	com.apple.CoreAuthUI
12	2021-05-24 10:58:46	2021-05-24 10:58:56	2	10	/app/inFocus	com.apple.mobilesafari

Figure 5.7 – The query result shows application usage

	START	END	GMT OFFSET	USAGE IN SECONDS	ZSTREAMNAME	ZVALUESTRING
1	2021-05-24 08:38:13	2021-05-24 10:56:11	2	8278	/app/inFocus	us.zoom.videomeetings
2	2021-06-06 14:53:43	2021-06-06 17:07:49	2	8046	/app/inFocus	com.apple.mobilesafari
3	2021-05-26 16:25:19	2021-05-26 18:38:19	2	7980	/app/inFocus	us.zoom.videomeetings
4	2021-06-02 11:03:58	2021-06-02 12:31:20	2	5242	/app/inFocus	com.apple.mobilesafari
5	2021-05-25 14:57:21	2021-05-25 16:23:48	2	5187	/app/inFocus	com.apple.mobilesafari
6	2021-05-24 13:37:35	2021-05-24 14:58:41	2	4866	/app/inFocus	com.apple.mobilesafari
7	2021-06-13 18:02:54	2021-06-13 19:23:42	2	4848	/app/inFocus	com.apple.mobilesafari
8	2021-06-13 16:44:36	2021-06-13 18:02:54	2	4698	/app/inFocus	com.apple.mobilesafari
9	2021-05-29 10:52:03	2021-05-29 12:09:21	2	4638	/app/inFocus	com.google.ios.youtube
10	2021-05-31 17:47:34	2021-05-31 19:04:05	2	4591	/app/inFocus	us.zoom.videomeetings
11	2021-06-01 22:03:57	2021-06-01 23:19:53	2	4556	/app/inFocus	com.netflix.Netflix
12	2021-06-06 11:15:07	2021-06-06 12:30:11	2	4504	/app/inFocus	com.apple.mobilesafari

Figure 5.8 – Analyzing long tasks

	START	END	GMT OFFSET	USAGE IN SECONDS	ZSTREAMNAME	ZSTRUCTUREDMETADATA
123	2021-06-01 22:03:04	2021-06-01 22:03:58	2	54	/media/nowPlaying	5149
124	2021-06-01 22:03:39	2021-06-01 22:03:57	2	18	/app/inFocus	428
125	2021-06-01 22:03:39	2021-06-01 23:19:53	2	4574	/app/usage	15
126	2021-06-01 22:03:40	2021-06-01 22:03:56	2	16	/display/orientation	NULL
127	2021-06-01 22:03:57	2021-06-01 23:19:53	2	4556	/app/inFocus	NULL
128	2021-06-01 22:03:58	2021-06-01 22:03:59	2	1	/media/nowPlaying	51731
129	2021-06-01 22:03:59	2021-06-01 22:04:00	2	1	/media/nowPlaying	51732
130	2021-06-01 22:04:00	2021-06-01 22:04:01	2	1	/media/nowPlaying	51733
131	2021-06-01 22:04:01	2021-06-01 23:19:29	2	4528	/media/nowPlaying	51732
132	2021-06-01 22:04:12	2021-06-01 23:19:48	2	4536	/display/orientation	NULL
133	2021-06-01 22:06:20	2021-06-01 22:08:20	2	120	/device/batteryPercentage	NULL
134	2021-06-01 22:07:00	2021-06-01 22:07:00	2	0	/dasd/batterytemperature	NULL

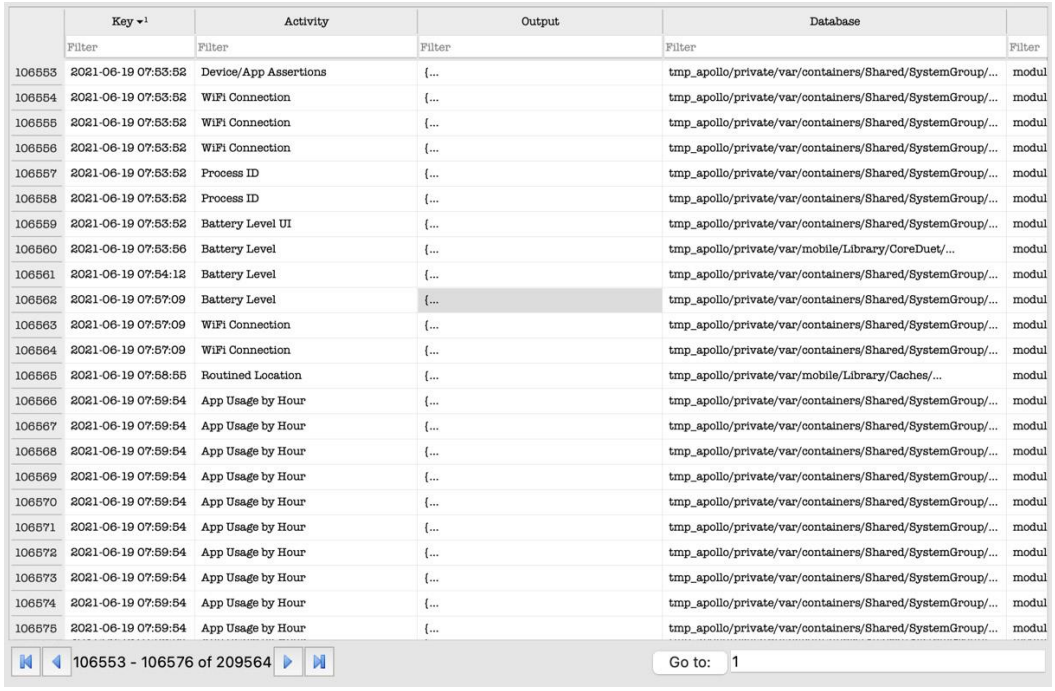
Figure 5.9 – The query result shows user interaction in the time span of interest

	ID	Z_DKNOWPLAYINGMETADATAKEY__TITLE
1	51732	Netflix - Quo Vado?

Figure 5.10 – The metadata associated with the /media/nowPlaying event

	DATE / TIME	GMT OFFSET	IS LOCKED
1	2021-05-23 14:49:13	2	LOCKED
2	2021-05-23 15:41:55	2	UNLOCKED
3	2021-05-23 15:58:59	2	LOCKED
4	2021-05-23 16:05:33	2	UNLOCKED
5	2021-05-23 16:52:36	2	LOCKED
6	2021-05-23 22:11:48	2	UNLOCKED
7	2021-05-23 22:50:47	2	LOCKED
8	2021-05-23 22:54:20	2	UNLOCKED
9	2021-05-23 23:23:24	2	LOCKED
10	2021-05-23 23:33:14	2	UNLOCKED
11	2021-05-24 00:08:13	2	LOCKED
12	2021-05-24 00:10:48	2	UNLOCKED

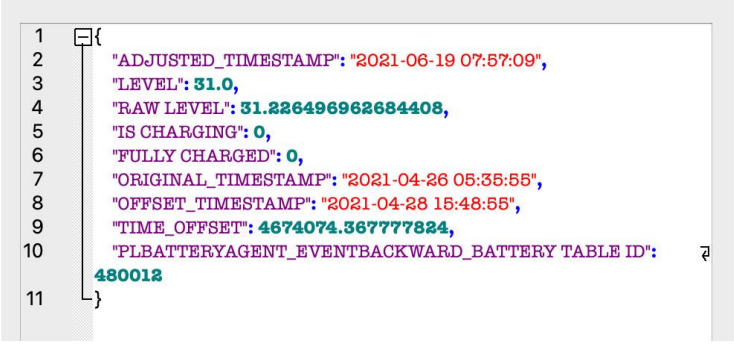
Figure 5.11 – The query shows the device lock status



	Key	Activity	Output	Database	
	Filter	Filter	Filter	Filter	Filter
106553	2021-06-19 07:53:52	Device/App Assertions	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106554	2021-06-19 07:53:52	WiFi Connection	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106555	2021-06-19 07:53:52	WiFi Connection	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106556	2021-06-19 07:53:52	WiFi Connection	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106557	2021-06-19 07:53:52	Process ID	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106558	2021-06-19 07:53:52	Process ID	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106559	2021-06-19 07:53:52	Battery Level UI	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106560	2021-06-19 07:53:56	Battery Level	{...	tmp_apollo/private/var/mobile/Library/CoreDuet/...	modul
106561	2021-06-19 07:54:12	Battery Level	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106562	2021-06-19 07:57:09	Battery Level	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106563	2021-06-19 07:57:09	WiFi Connection	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106564	2021-06-19 07:57:09	WiFi Connection	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106565	2021-06-19 07:58:55	Routined Location	{...	tmp_apollo/private/var/mobile/Library/Caches/...	modul
106566	2021-06-19 07:59:54	App Usage by Hour	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106567	2021-06-19 07:59:54	App Usage by Hour	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106568	2021-06-19 07:59:54	App Usage by Hour	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106569	2021-06-19 07:59:54	App Usage by Hour	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106570	2021-06-19 07:59:54	App Usage by Hour	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106571	2021-06-19 07:59:54	App Usage by Hour	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106572	2021-06-19 07:59:54	App Usage by Hour	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106573	2021-06-19 07:59:54	App Usage by Hour	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106574	2021-06-19 07:59:54	App Usage by Hour	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul
106575	2021-06-19 07:59:54	App Usage by Hour	{...	tmp_apollo/private/var/containers/Shared/SystemGroup/...	modul

106553 - 106576 of 209564 Go to: 1

Figure 5.12 – The Apollo database displayed in DB Browser




```
{
  "ADJUSTED_TIMESTAMP": "2021-06-19 07:57:09",
  "LEVEL": 31.0,
  "RAW LEVEL": 31.226496962684408,
  "IS CHARGING": 0,
  "FULLY CHARGED": 0,
  "ORIGINAL_TIMESTAMP": "2021-04-26 05:35:55",
  "OFFSET_TIMESTAMP": "2021-04-28 15:48:55",
  "TIME_OFFSET": 4674074.367777824,
  "PLBATTERYAGENT_EVENTBACKWARD_BATTERY TABLE ID": 480012
}
```


Figure 5.13 – The JSON object stored in Apollo's Output column

Code and Command

Command 5.1

```
SELECT id, uuid, datetime(timestamp, 'unixepoch') AS date
FROM Events;
```

Command 5.2:

```
SELECT id, uuid, datetime(timestamp, 'unixepoch',
'localtime') AS date FROM Events;
```

Command 5.3:

```
SELECT id, uuid, datetime(timestamp + 978307200,
'unixepoch', 'localtime') AS date FROM Events;
```

Command 5.4:

```
SELECT DISTINCT Z object.ZSTREAMNAME FROM Z object ORDER BY
ZSTREAMNAME;
```

Command 5.5

```
SELECT

datetime(Z object.ZSTARTDATE+978307200,'UNIXEPOCH',
'LOCALTIME') as "START",

datetime(Z object.ZENDDATE+978307200,'UNIXEPOCH',
'LOCALTIME') as "END",

Z object.ZSECONDSFROMGMT/3600 AS "GMT OFFSET",

(Z object.ZENDDATE-Z object.ZSTARTDATE) as "USAGE IN
SECONDS",

Z object.ZSTREAMNAME,

Z object.ZVALUESTRING

FROM Z object

WHERE ZSTREAMNAME IS "/app/inFocus"

AND Z object.ZSTARTDATE > 643507200

AND Z object.ZSTARTDATE < 643680000

ORDER BY "START";
```

Command 5.6

```
SELECT

datetime(Z object.ZSTARTDATE+978307200,'UNIXEPOCH',
'LOCALTIME') as "START",

datetime(Z object.ZENDDATE+978307200,'UNIXEPOCH',
'LOCALTIME') as "END",

Z object.ZSECONDSFROMGMT/3600 AS "GMT OFFSET",

(Z object.ZENDDATE-Z object.ZSTARTDATE) as "USAGE IN
SECONDS",

Z object.ZSTREAMNAME,

Z object.ZVALUESTRING

FROM Z object

WHERE ZSTREAMNAME IS "/app/inFocus"

ORDER BY "USAGE IN SECONDS" DESC;
```

Command 5.7

```
SELECT

datetime(Z object.ZSTARTDATE+978307200,'UNIXEPOCH',
'LOCALTIME') as "START",

datetime(Z object.ZENDDATE+978307200,'UNIXEPOCH',
'LOCALTIME') as "END",

Z object.ZSECONDSFROMGMT/3600 AS "GMT OFFSET",

(Z object.ZENDDATE-Z object.ZSTARTDATE) as "USAGE IN
SECONDS",

Z object.ZSTREAMNAME,

Z object.ZSTRUCTUREDMETADATA,

Z object.ZVALUESTRING

FROM Z object

WHERE Z object.ZSTARTDATE > 644266800
```

```
AND Z object.ZSTARTDATE < 644277599

ORDER BY "START";
```

Command 5.8:

```
SELECT

Z_PK AS "ID",

Z_DKNOWPLAYINGMETADATAKEY__TITLE

FROM ZSTRUCTUREDMETADATA

WHERE "ID" = "51732";
```

Command 5.9:

```
SELECT

datetime(Z object.ZSTARTDATE+978307200,'UNIXEPOCH',
'LOCALTIME') as "DATE / TIME",

Z object.ZSECONDSFROMGMT/3600 AS "GMT OFFSET",

CASE Z object.ZVALUEINTEGER

         WHEN '0' THEN 'UNLOCKED'

         WHEN '1' THEN 'LOCKED'

      END "IS LOCKED"

FROM Z object

WHERE Z object.ZSTREAMNAME LIKE "/device/isLocked"

ORDER BY "DATE / TIME";
```

Command 5.10:

```
SELECT

datetime(Z object.ZSTARTDATE+978307200,'UNIXEPOCH',
'LOCALTIME') as "DATE / TIME",

Z object.ZSECONDSFROMGMT/3600 AS "GMT OFFSET"

FROM Z object
```

```
WHERE Z object.ZSTREAMNAME LIKE "/display/isBacklit"
AND Z object.ZVALUEINTEGER = '1'
ORDER BY "DATE / TIME";
```

Command 5.11:

```
python apollo.py -output sql modules/ fs-extraction/
```

Links

- Apollo's Github repository: https://github.com/mac4n6/APOLLO

Chapter 6

Images

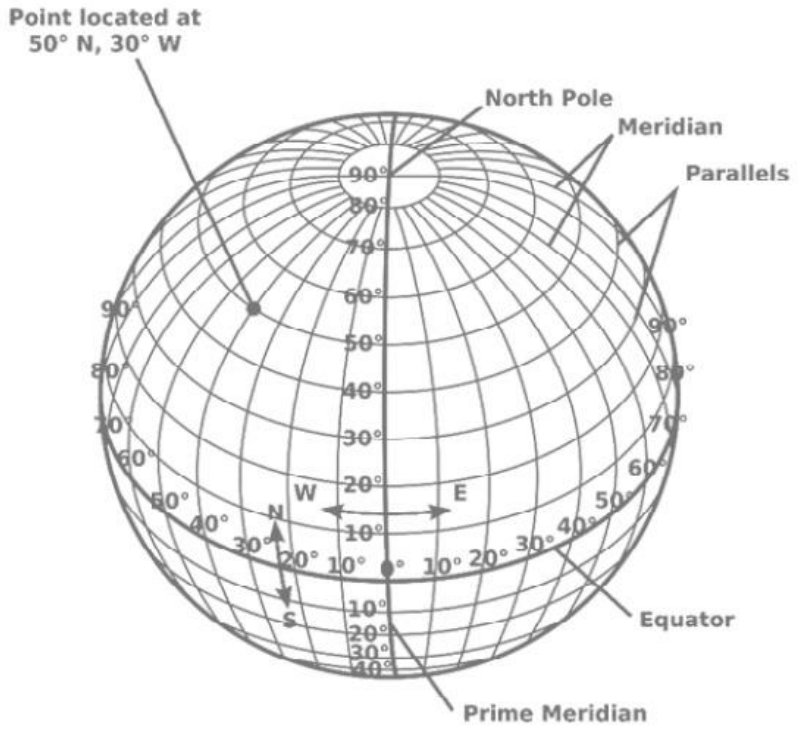


Figure 6.1 – Geographic locations are expressed using latitude and longitude

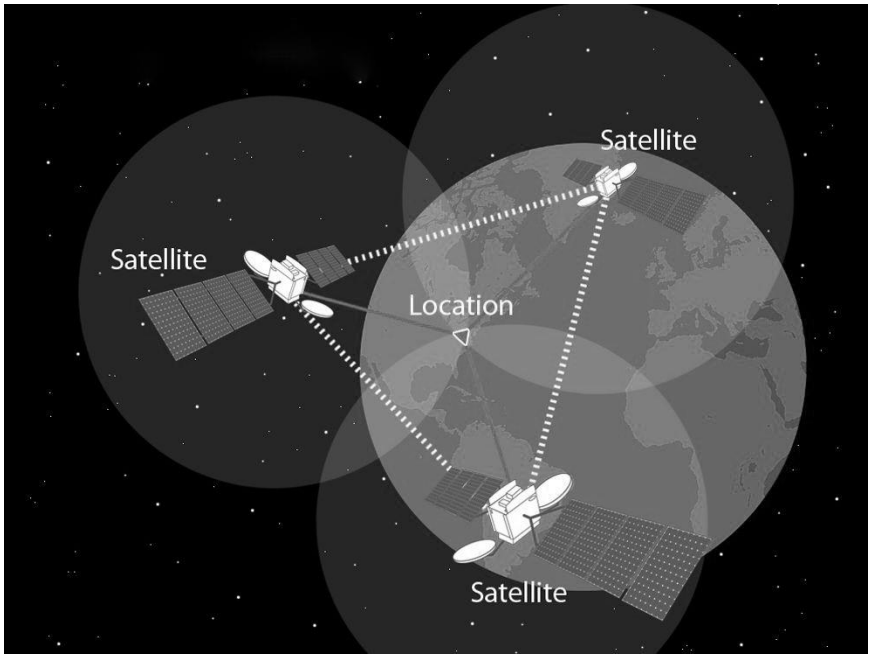


Figure 6.2 – GPS satellite triangulation calculates the receiver's location

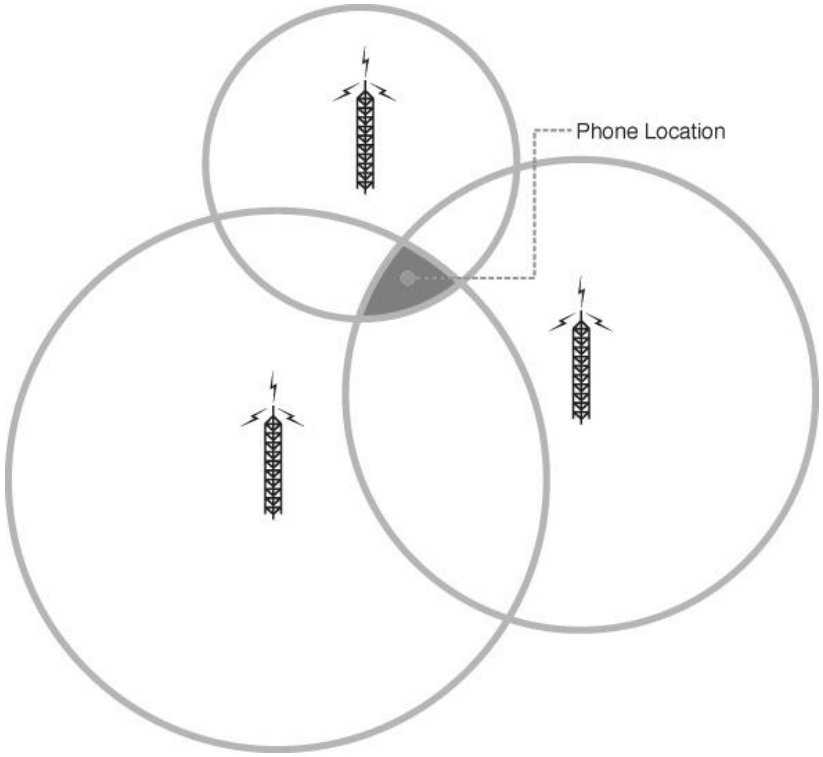


Figure 6.3 – Cell tower trilateration process

	TIMESTAMP	LATITUDE	LONGITUDE	ALTITUDE	SPEED (M/S)	SPEED (MPH)	SPEED (KMPH)	HORIZO
1	2020-04-12 08:28:48	35.6578612348278	-78.8705126020217	102.232818603516	-1.0	-2.23694	-3.6	
2	2020-04-12 08:28:49	35.6578694814717	-78.8705127743498	102.232818603516	-1.0	-2.23694	-3.6	
3	2020-04-12 08:28:51	35.6553155556039	-78.8704595665105	102.232818603516	-1.0	-2.23694	-3.6	19
4	2020-04-12 08:28:51	35.6513724593481	-78.8703774158301	102.232818603516	-1.0	-2.23694	-3.6	19
5	2020-04-12 08:28:51	35.653168602849	-78.8704148367899	102.232818603516	-1.0	-2.23694	-3.6	19
6	2020-04-12 08:32:23	35.6558862343305	-78.8704714569495	102.232818603516	-1.0	-2.23694	-3.6	
7	2020-04-12 08:32:24	35.6568810173752	-78.8704921817063	102.232818603516	-1.0	-2.23694	-3.6	
8	2020-04-12 08:32:22	35.6457080733981	-78.8702594108219	102.232818603516	-1.0	-2.23694	-3.6	
9	2020-04-12 08:33:54	35.6574534521418	-78.8705041068972	102.232818603516	-1.0	-2.23694	-3.6	
10	2020-04-12 08:35:33	35.6528454092581	-78.8704081031046	102.232818603516	-1.0	-2.23694	-3.6	19
11	2020-04-12 08:35:30	35.6575081235283	-78.8705052462567	102.232818603516	-1.0	-2.23694	-3.6	
12	2020-04-12 08:35:30	35.6571394683115	-78.8704975656647	102.232818603516	-1.0	-2.23694	-3.6	

Figure 6.4 – The query results show the device's location

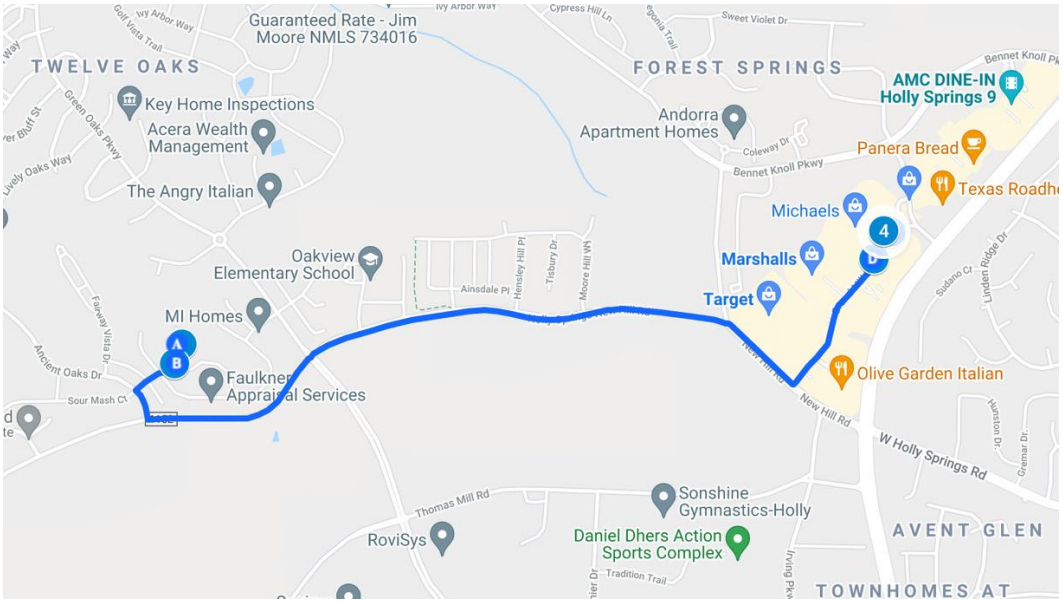


Figure 6.5 – The query results are mapped out to determine the user's journey

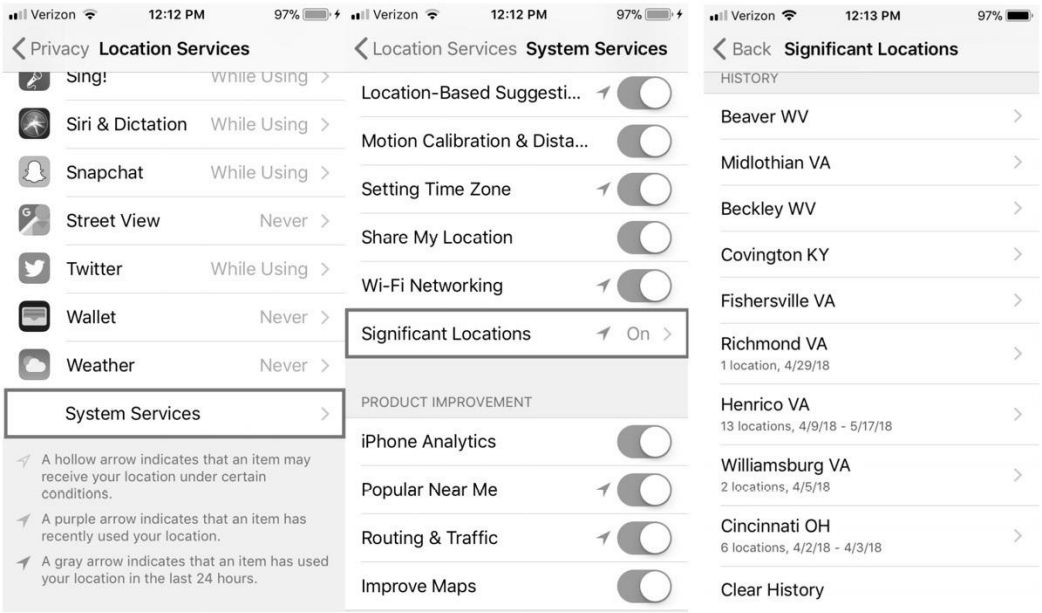


Figure 6.6– Significant Locations can be viewed directly on the device

	ENTRY	EXIT	DURATION (MINUTES)	LATITUDE	LONGITUDE	LOCATION UNCERTAINTY	DATA POINTS
1	2020-04-15 16:32:51	2020-04-16 23:06:51	1834.0	35.6593609837169	-78.8730660027371	32.8602404963937	7365
2	2020-04-17 19:35:14	2020-04-19 18:50:29	2835.25	35.6593609837169	-78.8730660027371	19.6819985540774	11438
3	2020-04-15 16:10:47	2020-04-15 16:20:19	9.5366063396136	35.6638133919877	-78.8479745012675	216.865382961763	39
4	2020-04-12 14:30:27	2020-04-15 16:03:04	4412.61572608352	35.6593609837169	-78.8730660027371	31.9387684448737	17726
5	2020-04-12 08:41:27	2020-04-12 13:55:27	314.0	35.6593609837169	-78.8730660027371	21.0690629834767	1257
6	2020-04-12 14:02:27	2020-04-12 14:19:27	17.0	35.6715623958704	-78.8770194584023	21.8376117000028	69

Figure 6.7 – The query results show details about significant location visits

	DATE	MAC ADDRESS	CHANNEL	RSSI
1	2020-04-12 12:46:19	f8:bb:bf:1e:fa:ea	161	-57
2	2020-04-12 12:46:19	f8:bb:bf:1e:fa:e4	161	-57
3	2020-04-12 12:46:20	80:da:13:72:52:64	149	-90
4	2020-04-12 12:46:20	f8:bb:bf:1e:fa:eb	48	-51
5	2020-04-12 12:46:20	58:ef:68:25:35:34	48	-90
6	2020-04-12 12:46:20	d8:d7:75:b6:28:9	48	-89
7	2020-04-12 12:46:20	80:da:13:72:52:6a	149	-89
8	2020-04-12 12:46:20	f8:bb:bf:8d:b9:c2	48	-78
9	2020-04-12 12:46:20	f8:bb:bf:90:a8:f2	48	-70
10	2020-04-12 12:46:20	b8:66:85:53:d3:7	44	-85
11	2020-04-12 12:46:20	7c:db:98:c2:30:34	40	-89
12	2020-04-12 12:46:20	f8:bb:bf:1e:fa:e8	48	-50

Figure 6.8 – The results show details about wireless devices that were scanned by the device

	TIMESTAMP	MCC	MNC	CI	HORIZONTAL ACCURACY	LATITUDE	LONGITUDE
1	2020-12-12 22:25:08	310	260	18005506	2142.0	35.6559143	-78.87465667
2	2020-12-12 22:25:08	310	260	256947784	1414.0	35.62225341	-78.87554168
3	2020-12-12 22:25:08	310	260	256769850	1414.0	35.66263961	-78.87255859
4	2020-12-12 22:25:08	311	480	39694094	2977.0	35.62737655	-78.84117126
5	2020-12-12 22:25:08	311	480	39694096	1414.0	35.62499618	-78.84272003
6	2020-12-12 22:25:08	310	410	196794049	2094.0	35.66758346	-78.87165069
7	2020-12-12 22:25:08	310	410	195403799	1414.0	35.64883804	-78.8419876
8	2020-12-12 22:25:08	311	480	116500503	1414.0	35.63685607	-78.86347198
9	2020-12-12 22:25:08	311	480	116500504	1414.0	35.63601303	-78.86531829
10	2020-12-12 22:25:08	310	410	195138569	1414.0	35.63536071	-78.86586761
11	2020-12-12 22:25:08	310	410	195001864	1414.0	35.66678237	-78.87182617
12	2020-12-12 22:25:08	311	480	39694114	13696.0	35.63697433	-78.86968231

Figure 6.9 – The query results show details about cell towers and geolocation

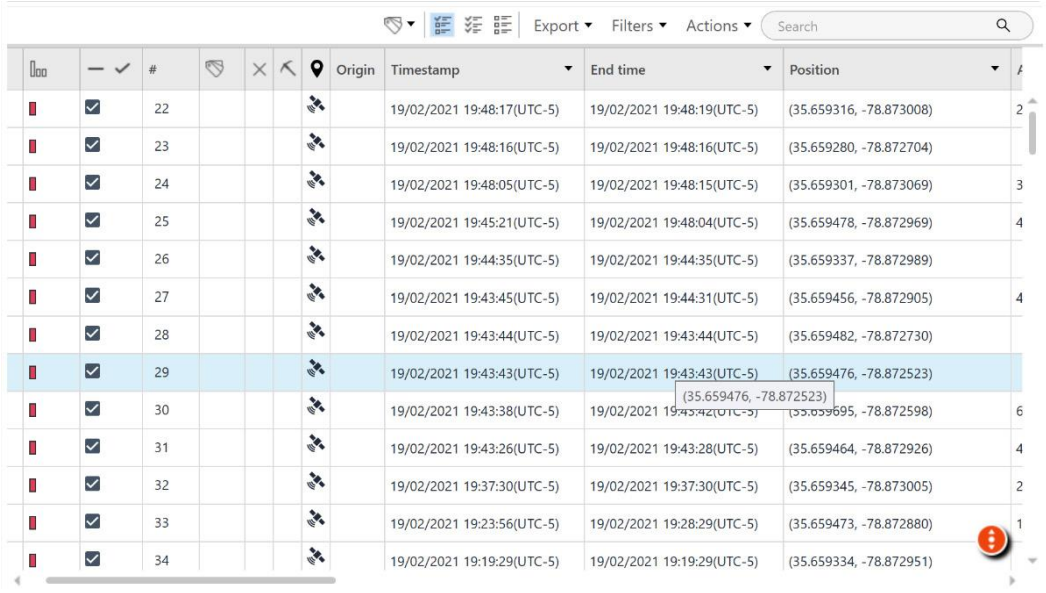


Figure 6.10 – Viewing location data in Physical Analyzer

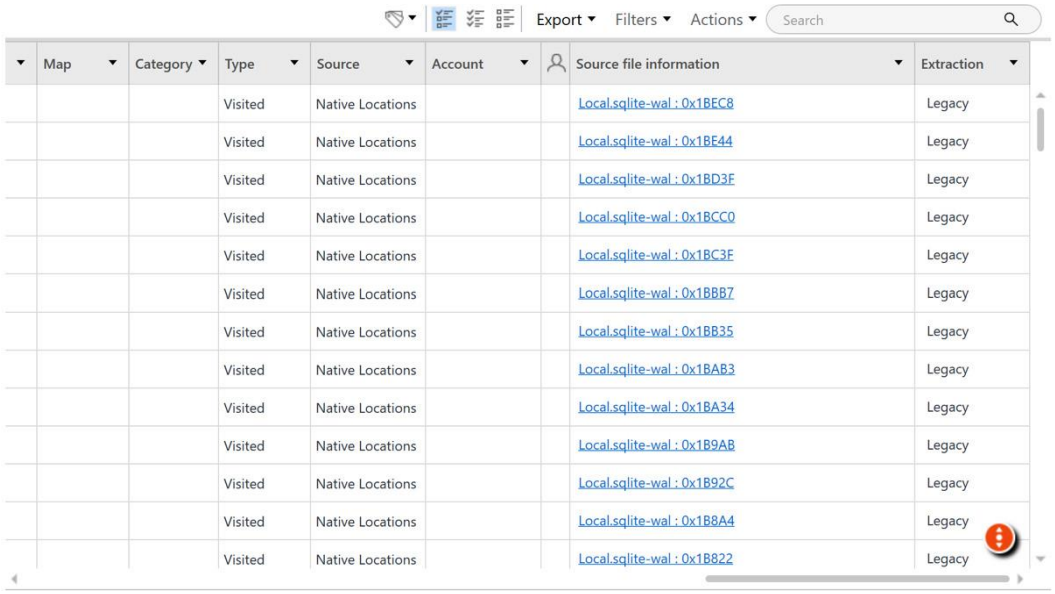


Figure 6.11 – Physical Analyzer specifies the source file for the artifact

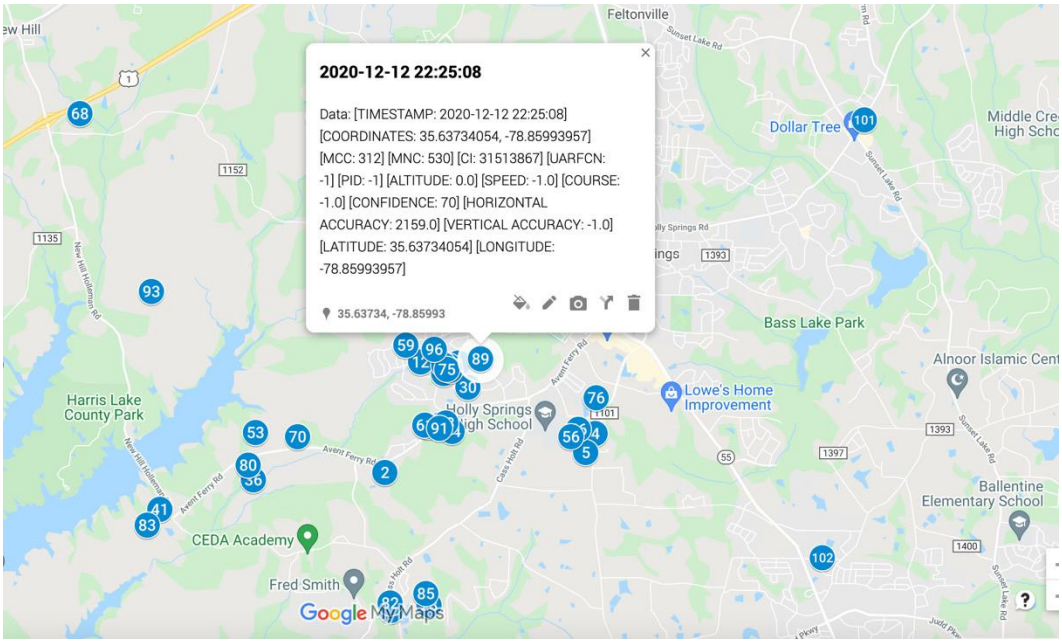


Figure 6.12 – Google Maps displaying location data exported by Apollo

Code and Command

Command 6.1:

```
SELECT

      DATETIME(ZTIMESTAMP + 978307200, 'UNIXEPOCH') AS
"TIMESTAMP",

      ZLATITUDE AS "LATITUDE",

      ZLONGITUDE AS "LONGITUDE",

      ZALTITUDE AS "ALTITUDE",

      ZSPEED AS "SPEED (M/S)",

      ZSPEED*2.23694 AS "SPEED (MPH)",

      ZSPEED*3.6 AS "SPEED (KMPH)",

      ZHORIZONTALACCURACY AS "HORIZONTAL ACCURACY",

      ZVERTICALACCURACY AS "VERTICAL ACCURACY"

   FROM

      ZRTCLLOCATIONMO;
```

Command 6.2:

```
SELECT

        DATETIME(ZRTLEARNEDLOCATIONOFINTERESTVISITMO.ZENTRY
DATE + 978307200, 'UNIXEPOCH') AS "ENTRY",

        DATETIME(ZRTLEARNEDLOCATIONOFINTERESTVISITMO.ZEXITD
ATE + 978307200, 'UNIXEPOCH') AS "EXIT",

        (ZRTLEARNEDLOCATIONOFINTERESTVISITMO.ZEXITDATE-
ZRTLEARNEDLOCATIONOFINTERESTVISITMO.ZENTRYDATE)/60.00 AS
"DURATION (MINUTES)",

        ZRTLEARNEDLOCATIONOFINTERESTMO.ZLOCATIONLATITUDE AS
"LATITUDE",
```

```
        ZRTLEARNEDLOCATIONOFINTERESTMO.ZLOCATIONLONGITUDE
AS "LONGITUDE",

        ZRTLEARNEDLOCATIONOFINTERESTVISITMO.ZLOCATIONUNCERT
AINTY AS "LOCATION UNCERTAINTY",

        ZRTLEARNEDLOCATIONOFINTERESTVISITMO.ZDATAPOINTCOUNT
AS "DATA POINTS"

     FROM

        ZRTLEARNEDLOCATIONOFINTERESTVISITMO

        LEFT JOIN

           ZRTLEARNEDLOCATIONOFINTERESTMO

           ON ZRTLEARNEDLOCATIONOFINTERESTMO.Z_PK =
ZRTLEARNEDLOCATIONOFINTERESTVISITMO.ZLOCATIONOFINTEREST;
```

Command 6.3:

```
SELECT

        DATETIME(ZRTWIFIACCESSPOINTMO.ZDATE + 978307200,
'UNIXEPOCH') AS "DATE",

        ZRTWIFIACCESSPOINTMO.ZMAC AS "MAC ADDRESS",

        ZRTWIFIACCESSPOINTMO.ZCHANNEL AS "CHANNEL",

        ZRTWIFIACCESSPOINTMO.ZRSSI AS "RSSI"

        FROM ZRTWIFIACCESSPOINTMO

        ORDER BY "DATE" ASC;
```

Command 6.4:

```
SELECT

           DATETIME(TIMESTAMP + 978307200,'UNIXEPOCH') AS
"TIMESTAMP",

           MCC AS "MCC",

           MNC AS "MNC",
```

```
            CI AS "CI",
            HORIZONTALACCURACY AS "HORIZONTAL ACCURACY",
            LATITUDE AS "LATITUDE",
            LONGITUDE AS "LONGITUDE"
     FROM LTECELLLOCATION;
```

Command 6.5:

```
python3 apollo.py extract -o sql -p apple -v 14 -k modules/
fs-extraction/
```

Command 6.6:

```
SELECT
            DATETIME(TIMESTAMP + 978307200,'UNIXEPOCH') AS
"TIMESTAMP",
            MAC AS "MAC",
            CHANNEL AS "CHANNEL",
            SCORE AS "SCORE",
            REACH AS "REACH",
            HORIZONTALACCURACY AS "HORIZONTAL ACCURACY",
            LATITUDE AS "LATITUDE",
            LONGITUDE AS "LONGITUDE"
     FROM WIFILOCATION;
```

Links

- Apollo's Github repository: https://github.com/mac4n6/APOLLO

Chapter 7

Images

com.apple.commcenter.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>CarrierBundleName</key>
        <string>310260_GID1-4276</string>
        <key>EnhancedLQMDataStallRegistration</key>
        <false/>
        <key>EnhancedLQMDataTransferTimeRegistration</key>
        <false/>
        <key>EnhancedLQMLinkPowerCostRegistration</key>
        <false/>
        <key>EnhancedLQMLinkQualityFingerPrintRegistration</key>
        <true/>
        <key>EnhancedLQMLinkStateRegistration</key>
        <true/>
        <key>EnhancedLQMTrafficClassRegistration</key>
        <true/>
        <key>LASDNextUpdate</key>
        <date>2020-12-23T23:04:32Z</date>
        <key>LastKnownICCID</key>
        <string>[redacted]</string>
        <key>LastKnownServingMcc</key>
        <integer>310</integer>
        <key>LastKnownServingMnc</key>
        <integer>260</integer>
        <key>NetworkPhoneNumber</key>
        <string>[redacted]</string>
        <key>NetworkPhoneNumberICCID</key>
        <string>[redacted]</string>
        <key>PersonalWallet</key>
        <dict>
                <key>[redacted]</key>
```

Figure 7.1 – The com.apple.commcenter PLIST

	TIMESTAMP	SERVICE	OPERATOR	STATUS
3	2020-05-08 15:25:25	None	*NULL*	NotRegistered
4	2020-08-24 16:55:03	None	*NULL*	NotRegistered
5	2020-12-12 22:19:12	None	*NULL*	NotRegistered
6	2020-12-12 22:21:15	None	*NULL*	NotRegistered
7	2020-12-12 22:30:00	None	*NULL*	Searching
8	2020-12-12 22:30:11	None	*NULL*	Emergency Only
9	2020-12-12 22:30:49	None	*NULL*	Denied
10	2020-12-12 22:30:49	None	*NULL*	Emergency Only
11	2020-12-12 22:33:38	None	Google Fi	RegisteredHome
12	2020-12-12 22:33:38	4G	Google Fi	RegisteredHome
13	2020-12-12 22:33:43	None	Google Fi	RegisteredHome
14	2020-12-12 22:33:43	4G	Google Fi	RegisteredHome

Figure 7.2 – Records of cellular registration

	First Name	Last Name	Contacts
1	Josh	Hickman	(919) 579-0479,(919) 391-2507,joshua.hickman1@me.com
2	This Is	DFIR	*NULL*

Figure 7.3 – Records extracted from the address book

	Date/Time	Phone Number	Location	Call in Seconds	Call Direction	Call Status	Service Provider
26	2020-04-10 16:19:47	+14082560700	San Jose, CA	0.0	Incoming	Call missed	com.apple.Telephony
27	2020-04-10 16:32:21	+14082560700	San Jose, CA	0.0	Incoming	Call missed	com.apple.Telephony
28	2020-04-10 16:40:42	+14082560700	San Jose, CA	0.0	Incoming	Call missed	com.apple.Telephony
29	2020-04-12 03:13:23	+17042751134	Fairview, NC	0.0	Outgoing	*NULL*	com.apple.Telephony
30	2020-04-12 03:14:46	+17042751134	Fairview, NC	12.2448190450668	Incoming	Call answered	com.apple.Telephony
31	2020-04-12 16:04:06	9192853680	Fuquay-Varina, NC	64.6294050216675	Outgoing	*NULL*	com.apple.Telephony
32	2020-04-12 17:26:43	joshua.hickman1@me.com	<<RecentsNumberLocati...	98.0257830619812	Outgoing	*NULL*	com.apple.FaceTime
33	2020-04-13 20:41:27	+14082560700	San Jose, CA	0.0	Incoming	Call missed	com.apple.Telephony
34	2020-04-13 23:57:05	+14082560700	San Jose, CA	0.0	Incoming	Call missed	com.apple.Telephony
35	2020-04-14 14:43:26	+14082560700	San Jose, CA	0.0	Incoming	Call missed	com.apple.Telephony
36	2020-04-14 17:25:18	+14082560700	San Jose, CA	0.0	Incoming	Call missed	com.apple.Telephony
37	2020-04-14 17:56:12	+14082560700	San Jose, CA	0.0	Incoming	Call missed	com.apple.Telephony

Figure 7.4 – Data extracted from the call log

Figure 7.5 – The com.apple.wifi.plist showing details of known networks

	PROCESS TIMESTAMP	PROCESS NAME	BUNDLE ID	WWAN IN	WWAN OUT
85	2020-03-22 15:14:45	com.apple.WebKit/imgurmobile	imgurmobile	3102060.0	275109.0
86	2020-03-22 15:16:40	mediaserverd/imgurmobile	imgurmobile	114007059.0	3366467.0
87	2020-03-22 15:10:24	nsurlsessiond/imgurmobile	imgurmobile	926878.0	45097.0
88	2020-03-22 18:02:57	nsurlsessiond/co.babypenguin.imo	co.babypenguin.imo	8978.0	6276.0
89	2020-03-22 18:59:22	mDNSResponder/com.burbn.instagram	com.burbn.instagram	5521.0	3440.0
90	2020-03-22 19:12:02	Instagram/com.burbn.instagram	com.burbn.instagram	15820543.0	662197.0
91	2020-03-22 18:48:48	InstagramNotific/com.burbn.instagram	com.burbn.instagram	28270.0	20203.0
92	2020-03-25 02:00:51	Threads/com.burbn.threads	com.burbn.threads	6336.0	23244.0
93	2020-03-27 19:02:30	appstored/com.kik.chat	com.kik.chat	14819.0	5875.0
94	2020-03-30 16:08:45	mDNSResponder/com.belkin.plugin	com.belkin.plugin	1452.0	1645.0
95	2020-03-30 16:07:06	WeMo_Universal/com.belkin.plugin	com.belkin.plugin	14913.0	8634.0
96	2020-04-04 16:16:50	mDNSResponder/com.facebook.talk	com.facebook.talk	367.0	485.0

Figure 7.6 – The query results showing the processes running on the device and their data usage

	Start	End	Status	Address	Device
1	2020-03-27 19:32:42	2020-03-27 19:32:52	Connected	7C:04:D0:89:89:A0	Josh's AirPods
2	2020-03-27 19:32:52	2020-03-27 19:33:03	Disconnected	7C:04:D0:89:89:A0	Josh's AirPods
3	2020-03-27 19:33:03	2020-03-27 19:33:05	Connected	7C:04:D0:89:89:A0	Josh's AirPods
4	2020-03-27 19:33:05	2020-03-27 19:54:15	Disconnected	7C:04:D0:89:89:A0	Josh's AirPods
5	2020-03-27 19:54:16	2020-03-27 20:22:18	Connected	7C:04:D0:89:89:A0	Josh's AirPods
6	2020-03-27 20:22:18	2020-03-27 21:44:02	Disconnected	7C:04:D0:89:89:A0	Josh's AirPods
7	2020-03-27 21:44:04	2020-03-27 21:44:05	Connected	7C:04:D0:89:89:A0	Josh's AirPods
8	2020-03-27 21:44:05	2020-03-27 21:44:18	Disconnected	7C:04:D0:89:89:A0	Josh's AirPods
9	2020-03-27 21:44:20	2020-03-27 21:44:42	Connected	7C:04:D0:89:89:A0	Josh's AirPods
10	2020-03-27 21:44:42	2020-03-27 21:44:48	Disconnected	7C:04:D0:89:89:A0	Josh's AirPods
11	2020-03-27 21:44:49	2020-03-27 21:44:55	Connected	7C:04:D0:89:89:A0	Josh's AirPods

Figure 7.7 – The query displaying Bluetooth connection events

```
SELECT
    DATETIME(HISTORY_VISITS.VISIT_TIME+978307200,'UNIXEPOCH') AS "VISIT TIME",
    HISTORY_ITEMS.URL AS "URL",
    HISTORY_ITEMS.VISIT_COUNT AS "VISIT COUNT",
    HISTORY_VISITS.TITLE AS "TITLE",
    CASE HISTORY_VISITS.ORIGIN
      WHEN 1 THEN "ICLOUD SYNCED DEVICE"
      WHEN 0 THEN "VISITED FROM THIS DEVICE"
      ELSE HISTORY_VISITS.ORIGIN
    END "ICLOUD SYNC"
  FROM HISTORY_ITEMS
  LEFT OUTER JOIN HISTORY_VISITS ON HISTORY_ITEMS.ID == HISTORY_VISITS.HISTORY_ITEM;
```

	VISIT TIME	URL	VISIT COUNT	TITLE
1	2020-03-28 01:00:17	https://www.google.com/search?...	3	when does mlb start 2020 - Google Search
2	2020-03-28 00:58:37	https://www.google.com/search?...	3	when does mlb start 2020 - Google Search
3	2020-03-28 00:58:36	https://www.google.com/search?...	3	when does mlb start 2020 - Google Search
4	2020-03-28 00:59:32	https://www.google.com/amp/s/www.mlb.com/amp/news/...	3	MLB 2020 season delayed
5	2020-03-28 00:59:31	https://www.google.com/amp/s/www.mlb.com/amp/news/...	3	MLB 2020 season delayed
6	2020-03-28 00:59:03	https://www.google.com/amp/s/www.mlb.com/amp/news/...	3	MLB 2020 season delayed
7	2020-03-28 01:02:05	https://www.google.com/search?...	1	when does mlb start 2020 - Google Search
8	2020-03-28 01:02:44	https://www.google.com/search?...	2	Is the NHL going to resume? - Google Search
9	2020-03-28 01:02:44	https://www.google.com/search?...	2	Is the NHL going to resume? - Google Search
10	2020-03-28 01:03:53	https://www.nhl.com/news/commissioner-gary-bettman-...	2	NHL expects to resume season after pause for coronavirus

Figure 7.8 – Analyzing the browsing history of History.db

	DATE	TITLE	URL	GMT OFFSET
1	2020-03-28 00:58:40	when does mlb start 2020 - Google Search	https://www.google.com/search?...	-4
2	2020-03-28 01:02:50	Is the NHL going to resume? - Google Search	https://www.google.com/search?...	-4
3	2020-03-28 01:04:05	NHL expects to resume season after pause for ...	https://www.nhl.com/news/commissioner-gary-bettman-...	-4
4	2020-03-28 01:05:45	Is the NHL going to resume? - Google Search	https://www.google.com/search?...	-4
5	2020-03-28 01:06:35	Apple	https://www.apple.com/	-4
6	2020-03-28 01:07:45	iPad Pro - Apple	https://www.apple.com/ipad-pro/	-4
7	2020-03-28 01:37:10	iPad Pro - Apple	https://www.apple.com/ipad-pro/	-4
8	2020-03-28 01:38:55	Cult of Mac | Tech and culture through an Apple lens	https://www.cultofmac.com/	-4
9	2020-03-28 01:43:45	Apple	https://www.apple.com/	-4
10	2020-03-28 01:43:55	DFIR Review	https://dfir.pubpub.org/	-4

Figure 7.9 – Analyzing the browsing history of KnowledgeC.db

Code and Commands

Commands 7.1:

```
SELECT
            DATETIME(TIMESTAMP , 'UNIXEPOCH') AS TIMESTAMP,
```

```
            DATAIND AS "SERVICE",
            OPERATOR AS "OPERATOR",
            STATUS AS "STATUS"
        FROM PLBBAGENT_EVENTFORWARD_TELEPHONYREGISTRATION;
```

Commands 7.2:

```
SELECT p.First                AS "First Name",
       p.Last                 AS "Last Name",
       GROUP_CONCAT(c.value) AS "Contacts"
FROM   ABPerson AS p
       LEFT JOIN ABMultiValue AS c
              ON c.record_id = p.rowid
GROUP  BY p.rowid,
          p.first;
```

Commands 7.3:

```
SELECT
DATETIME(ZDATE + 978307200, 'unixepoch', 'localtime') AS
"Date/Time",
ZADDRESS AS "Phone Number",
ZLOCATION AS "Location",
ZDURATION AS "Call in Seconds",
CASE
WHEN ZORIGINATED = 0 THEN "Incoming"
     WHEN ZORIGINATED = 1 THEN "Outgoing"
END AS "Call Direction",
CASE
```

```
     WHEN ZANSWERED = 0 AND ZORIGINATED = 0 THEN "Call
missed"

     WHEN ZANSWERED = 1 AND ZORIGINATED = 0 THEN "Call
answered"

END AS "Call Status",

ZSERVICE_PROVIDER AS "Service Provider"

FROM ZCALLRECORD;
```

Command 7.4:

```
SELECT

           DATETIME(ZPROCESS.ZTIMESTAMP+ 978307200,
'UNIXEPOCH') AS "PROCESS TIMESTAMP",

           ZPROCESS.ZPROCNAME AS "PROCESS NAME",

           ZPROCESS.ZBUNDLENAME AS "BUNDLE ID",

           ZLIVEUSAGE.ZWWANIN AS "WWAN IN",

           ZLIVEUSAGE.ZWWANOUT AS "WWAN OUT"

     FROM ZLIVEUSAGE

     LEFT JOIN ZPROCESS ON ZLIVEUSAGE.ZHASPROCESS =
ZPROCESS.Z_PK;
```

Command 7.5:

```
SELECT

DATETIME(ZOBJECT.ZSTARTDATE + 978307200, 'unixepoch',
'localtime') AS "Start",

DATETIME(ZOBJECT.ZENDDATE + 978307200, 'unixepoch',
'localtime') AS "End",

CASE

     WHEN ZOBJECT.ZVALUEINTEGER = 0 THEN "Disconnected"

     WHEN ZOBJECT.ZVALUEINTEGER = 1 THEN "Connected"
```

```
END AS "Status",

ZSTRUCTUREDMETADATA.Z_DKBLUETOOTHMETADATAKEY__ADDRESS AS
"Address",

ZSTRUCTUREDMETADATA.Z_DKBLUETOOTHMETADATAKEY__NAME AS
"Device"

FROM ZSTRUCTUREDMETADATA

LEFT JOIN ZOBJECT ON ZSTRUCTUREDMETADATA.Z_PK =
ZOBJECT.ZSTRUCTUREDMETADATA

WHERE ZOBJECT.ZSTREAMNAME = "/bluetooth/isConnected"

ORDER BY "Start" ASC;
```

Command 7.6:

```
SELECT

      DATETIME(ZOBJECT.ZSTARTDATE+978307200,'UNIXEPOCH') AS
"DATE",

      ZSTRUCTUREDMETADATA.Z_DKSAFARIHISTORYMETADATAKEY__TIT
LE AS "TITLE",

      ZOBJECT.ZVALUESTRING AS "URL",

      ZOBJECT.ZSECONDSFROMGMT/3600 AS "GMT OFFSET"

     FROM ZOBJECT

            LEFT JOIN

         ZSTRUCTUREDMETADATA

         ON ZOBJECT.ZSTRUCTUREDMETADATA =
ZSTRUCTUREDMETADATA.Z_PK

      LEFT JOIN

         ZSOURCE

         ON ZOBJECT.ZSOURCE = ZSOURCE.Z_PK

   WHERE
```

```
      ZSTREAMNAME IS "/safari/history"
ORDER BY DATE ASC;
```

Links

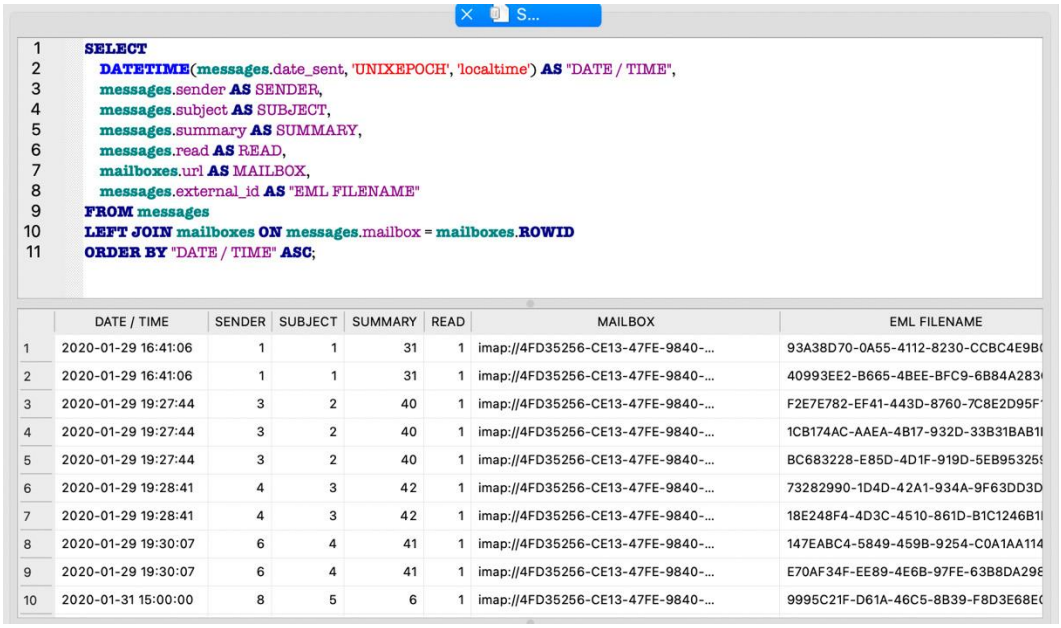
- Safari cookies file format specification: https://github.com/libyal/dtformats/blob/main/documentation/Safari%20Cookies.asciidoc
- Safari-Binary-Cookie-Parser Python script by Mari DeGrazia: https://github.com/mdegrazia/Safari-Binary-Cookie-Parser

Chapter 8

Images

	ROWID	url
	Filter	Filter
1	1	imap://4FD35256-CE13-47FE-9840-EBEB5B9FD9C1/INBOX
2	2	imap://4FD35256-CE13-47FE-9840-EBEB5B9FD9C1/%5BGmail%5D/%5CImportant
3	3	imap://4FD35256-CE13-47FE-9840-EBEB5B9FD9C1/%5BGmail%5D/%5CAllMail
4	4	imap://4FD35256-CE13-47FE-9840-EBEB5B9FD9C1/%5BGmail%5D/%5CSpam
5	5	imap://4FD35256-CE13-47FE-9840-EBEB5B9FD9C1/%5BGmail%5D/%5CSent
6	6	local://LocalAccountId/x-apple-transient-drafts
7	7	local://LocalAccountId/Outbox
8	8	imap://4FD35256-CE13-47FE-9840-EBEB5B9FD9C1/%5BGmail%5D/%5CTrash
9	9	imap://4FD35256-CE13-47FE-9840-EBEB5B9FD9C1/%5BGmail%5D/%5CStarred
10	10	imap://4FD35256-CE13-47FE-9840-EBEB5B9FD9C1/%5BGmail%5D/%5CDrafts

Figure 8.1 – The mailboxes table displayed in DB Browser



```
SELECT
  DATETIME(messages.date_sent, 'UNIXEPOCH', 'localtime') AS "DATE / TIME",
  messages.sender AS SENDER,
  messages.subject AS SUBJECT,
  messages.summary AS SUMMARY,
  messages.read AS READ,
  mailboxes.url AS MAILBOX,
  messages.external_id AS "EML FILENAME"
FROM messages
LEFT JOIN mailboxes ON messages.mailbox = mailboxes.ROWID
ORDER BY "DATE / TIME" ASC;
```

	DATE / TIME	SENDER	SUBJECT	SUMMARY	READ	MAILBOX	EML FILENAME
1	2020-01-29 16:41:06	1	1	31	1	imap://4FD35256-CE13-47FE-9840-...	93A38D70-0A55-4112-8230-CCBC4E9B(
2	2020-01-29 16:41:06	1	1	31	1	imap://4FD35256-CE13-47FE-9840-...	40993EE2-B665-4BEE-BFC9-6B84A283(
3	2020-01-29 19:27:44	3	2	40	1	imap://4FD35256-CE13-47FE-9840-...	F2E7E782-EF41-443D-8760-7C8E2D95F'
4	2020-01-29 19:27:44	3	2	40	1	imap://4FD35256-CE13-47FE-9840-...	1CB174AC-AAEA-4B17-932D-33B31BAB1I
5	2020-01-29 19:27:44	3	2	40	1	imap://4FD35256-CE13-47FE-9840-...	BC683228-E85D-4D1F-919D-5EB95325(
6	2020-01-29 19:28:41	4	3	42	1	imap://4FD35256-CE13-47FE-9840-...	73282990-1D4D-42A1-934A-9F63DD3D
7	2020-01-29 19:28:41	4	3	42	1	imap://4FD35256-CE13-47FE-9840-...	18E248F4-4D3C-4510-861D-B1C1246B1I
8	2020-01-29 19:30:07	6	4	41	1	imap://4FD35256-CE13-47FE-9840-...	147EABC4-5849-459B-9254-C0A1AA114
9	2020-01-29 19:30:07	6	4	41	1	imap://4FD35256-CE13-47FE-9840-...	E70AF34F-EE89-4E6B-97FE-63B8DA29E
10	2020-01-31 15:00:00	8	5	6	1	imap://4FD35256-CE13-47FE-9840-...	9995C21F-D61A-46C5-8B39-F8D3E68E(

Figure 8.2 – The SQL query extracts email metadata from the messages table

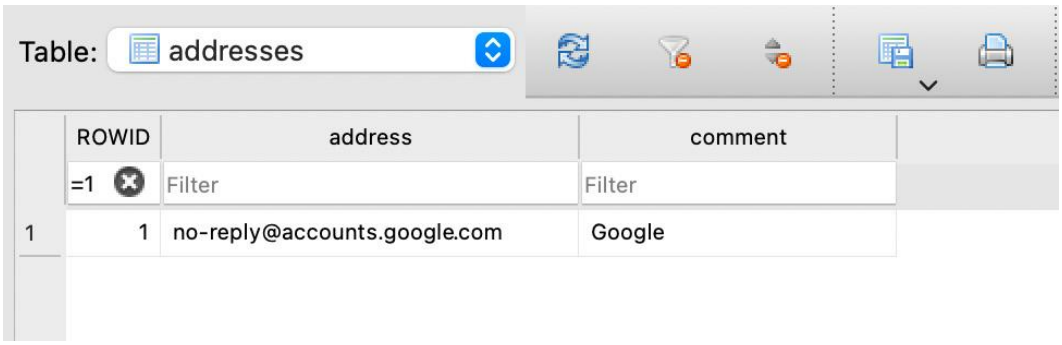


Table: addresses

	ROWID	address	comment
	=1	Filter	Filter
1	1	no-reply@accounts.google.com	Google

Figure 8.3 – The addresses table filtered by ROWID

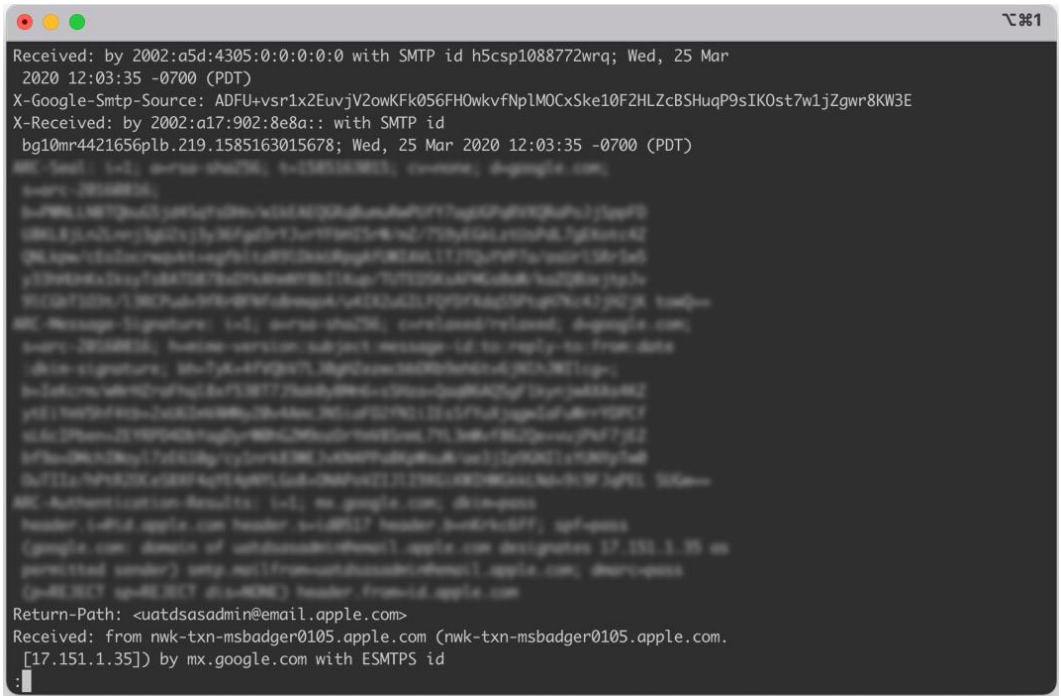


Figure 8.4 – An EML file viewed through the terminal showing the email's content

	Message Date	text	service	account	account_login
1	2020-03-22 02:10:59	G-773293 is your Google verification code....	SMS	p:+19195794674	P:+19195794674
2	2020-03-22 16:16:16	imo code: 258465...	SMS	p:+19195794674	P:+19195794674
3	2020-03-22 20:21:40	Please enter 737950 into LINE within the next 30 mins....	SMS	p:+19195794674	P:+19195794674
4	2020-03-23 02:06:38	Please enter this code in the app to reset your MeWe ...	SMS	p:+19195794674	P:+19195794674
5	2020-03-23 02:38:03	Your Signal verification code: 903-394...	SMS	p:+19195794674	P:+19195794674
6	2020-03-23 18:41:36	🔔 ThisIs,you have 2 new notifications on Facebook: ...	SMS	p:+19195794674	P:+19198887386
7	2020-03-24 02:21:34	Telegram code: 95037...	SMS	p:+19195794674	P:+19195794674
8	2020-03-24 02:27:40	[TikTok] 8494 is your verification code,valid for 5 minutes.	SMS	p:+19195794674	P:+19195794674
9	2020-03-24 02:52:23	Your WhatsApp code: 203-591...	SMS	p:+19195794674	P:+19195794674
10	2020-03-25 01:47:38	Your Skout verification code is 3079....	SMS	p:+19195794674	P:+19195794674

	account	account_login	chat_identifier	Date Read	Message Direction	Last Read
1	p:+19195794674	P:+19195794674	22000	2020-03-22 02:11:03	Incoming	2020-04-07 03:05:23
2	p:+19195794674	P:+19195794674	+18634205597	2020-03-22 16:16:19	Incoming	2020-03-22 16:16:19
3	p:+19195794674	P:+19195794674	+13176444906	2020-03-22 20:21:53	Incoming	2020-03-22 20:21:53
4	p:+19195794674	P:+19195794674	85760	2020-03-23 02:06:44	Incoming	2020-03-23 02:06:44
5	p:+19195794674	P:+19195794674	+17026604496	2020-03-23 02:38:05	Incoming	2020-03-23 02:38:05
6	p:+19195794674	P:+19198887386	32665	2020-03-24 02:00:59	Incoming	2020-04-16 00:14:06
7	p:+19195794674	P:+19195794674	+12074242620	2020-03-24 02:21:36	Incoming	2020-03-24 02:21:36
8	p:+19195794674	P:+19195794674	+13233208923	2020-03-24 02:27:42	Incoming	2020-03-24 02:27:42
9	p:+19195794674	P:+19195794674	47543	2020-03-24 02:52:25	Incoming	2020-03-24 02:52:25
10	p:+19195794674	P:+19195794674	+12349008110	2020-03-25 01:47:41	Incoming	2020-03-25 01:47:41

Figure 8.5 – Extracting message data from sms.db

	Z_PK	DATE	ZTEXT
1	1	2020-03-26 19:42:57	*NULL*
2	2	2020-03-26 19:42:57	What's up?!
3	3	2020-03-26 19:43:48	Not much. Just waiting to hop on a conference call. You?
4	4	2020-03-26 19:44:44	A little busy. Finished one report this morning and going to...
5	6	2020-03-26 19:47:12	Not yet but the effects of this will be felt later I'm sure.
6	7	2020-03-27 01:35:26	*NULL*
7	8	2020-03-27 01:36:36	Lol!!
8	9	2020-03-27 01:37:02	My turn.
9	10	2020-03-27 01:38:00	*NULL*
10	11	2020-04-11 20:23:43	*NULL*

Figure 8.6 – The Z_PK column stores a unique identifier for each message

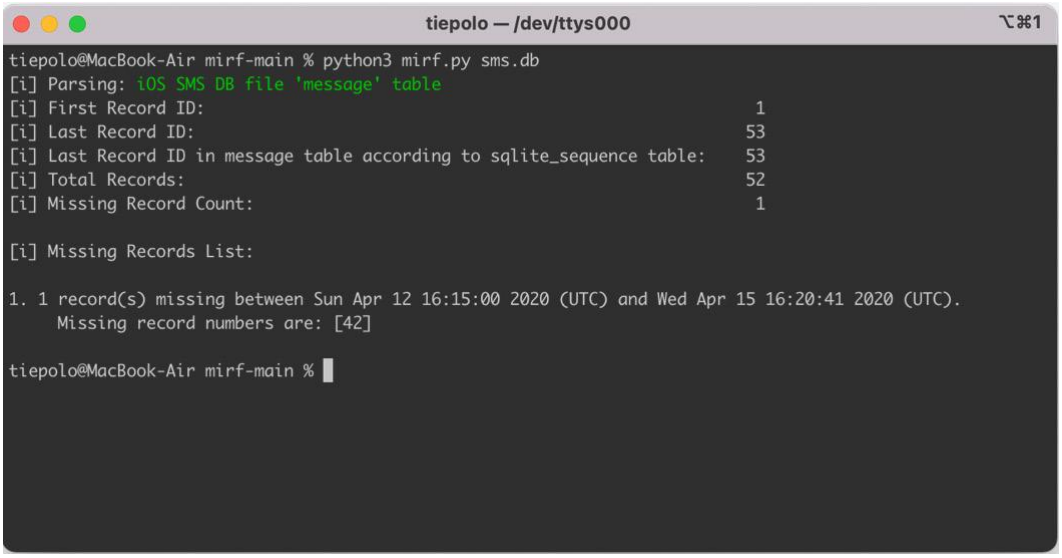


Figure 8.7 – Running Mirf to detect deleted messages

Code and Commands

Command 8.1:

```
SELECT

DATETIME(messages.date_sent, 'UNIXEPOCH', 'localtime') AS
"DATE / TIME",

messages.sender AS SENDER,

messages.subject AS SUBJECT,

messages.summary AS SUMMARY,

messages.read AS READ,

mailboxes.url AS MAILBOX,

messages.external_id AS "EML FILENAME"

FROM messages

LEFT JOIN mailboxes ON messages.mailbox = mailboxes.ROWID
```

```
ORDER BY "DATE / TIME" ASC;
```

Command 8.2:

```
SELECT

CASE when LENGTH(chat_message_join.message_date)=18 THEN

     datetime(chat_message_join.message_date/1000000000 +
978307200,'unixepoch','localtime')

     when LENGTH(chat_message_join.message_date)=9 THEN

     datetime(chat_message_join.message_date +
978307200,'unixepoch','localtime')

     else 'NA'

END as "Message Date",

message.text,

message.service,

message.account,

chat.account_login,

chat.chat_identifier,

CASE when LENGTH(message.date_read)=18 THEN

     datetime(message.date_read/1000000000 +
978307200,'unixepoch','localtime')

     when LENGTH(message.date_read)=9 THEN

     datetime(message.date_read+978307200,'unixepoch','loca
ltime')

     else 'NA'

END as "Date Read",

CASE when message.is_read=1

     THEN 'Incoming'

     when message.is_read=0
```

```
     THEN 'Outgoing'

END AS "Message Direction",

CASE when LENGTH(chat.last_read_message_timestamp)=18 THEN

     datetime(chat.last_read_message_timestamp/1000000000+9
78307200,'unixepoch','localtime')

     when LENGTH(chat.last_read_message_timestamp)=9 THEN

     datetime(chat.last_read_message_timestamp +
978307200,'unixepoch','localtime')

     else 'NA'

END as "Last Read"

FROM message

left join chat_message_join on
chat_message_join.message_id=message.ROWID

left join chat on chat.ROWID=chat_message_join.chat_id

left join attachment on
attachment.ROWID=chat_message_join.chat_id

order by "Message Date" ASC;
```

Command 8.3:

```
pip install -r requirements.txt
```

Chapter 9

Images

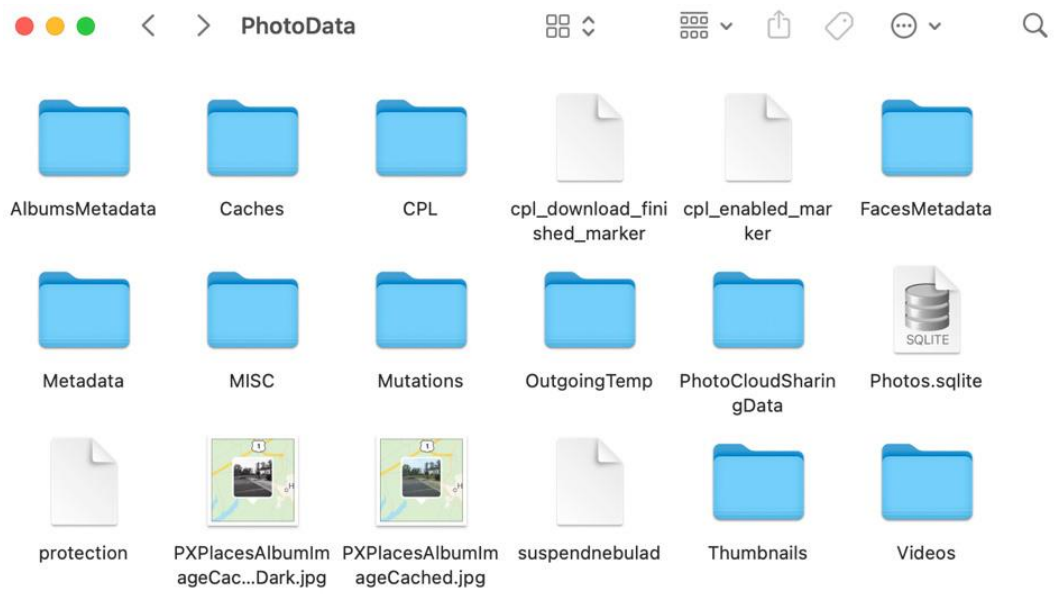


Figure 9.1 – The PhotoData folder contains media metadata

	Z_PK	Created date	Modified date	Deleted date	File name	Directory	Width	Height	Hidden
1	1	2020-03-22 13:05:24	2020-04-06 15:02:38	NULL	IMG_0001.JPG	DCIM/100APPLE	1125	1137	0
2	2	2020-03-22 13:05:39	2020-04-06 15:02:38	NULL	IMG_0002.JPG	DCIM/100APPLE	3550	3150	0
3	3	2020-03-22 13:10:50	2020-04-13 19:32:52	NULL	IMG_0003.JPG	DCIM/100APPLE	1668	860	0
4	4	2020-03-22 13:25:02	2020-04-06 15:02:38	NULL	IMG_0004.JPG	DCIM/100APPLE	1117	886	0
5	5	2020-03-22 13:25:02	2020-04-06 15:02:38	NULL	IMG_0005.JPG	DCIM/100APPLE	1522	1775	0
6	6	2020-03-22 13:25:02	2020-04-06 15:02:38	NULL	IMG_0006.JPG	DCIM/100APPLE	1170	1175	0
7	7	2020-03-22 13:25:02	2020-04-19 18:25:57	NULL	IMG_0007.JPG	DCIM/100APPLE	1178	781	0
8	8	2020-03-22 13:25:02	2020-04-13 19:32:52	NULL	IMG_0008.JPG	DCIM/100APPLE	1125	1156	0
9	9	2020-03-22 13:31:03	2020-04-06 15:02:38	NULL	IMG_0009.JPG	DCIM/100APPLE	828	637	0
10	10	2020-03-22 13:40:14	2020-04-19 18:25:58	NULL	IMG_0010.JPG	DCIM/100APPLE	620	553	0

Figure 9.2 – The results of the query from the Photos.sqlite database

```
SELECT ZGENERICASSET.Z_PK AS 'GenericAsset_zpk',
 ZGENERICASSET.ZADDITIONALATTRIBUTES AS 'AddAttributes_Key',
 ZDETECTEDFACE.ZASSET AS 'DetectedFaceAsset',
 CASE ZGENERICASSET.ZKIND
  WHEN 0 THEN 'Photo'
  WHEN 1 THEN 'Video'
 END AS 'Kind',
 DateTime(ZGENERICASSET.ZDATECREATED + 978307200, 'UNIXEPOCH') AS 'DateCreated',
 ZADDITIONALASSETATTRIBUTES.ZEXIFTIMESTAMPSTRING AS 'EXIFtimestamp',
 DateTime(ZADDITIONALASSETATTRIBUTES.ZSCENEANALYSISTIMESTAMP + 978307200, 'UNIXEPOCH') A
 DateTime(ZGENERICASSET.ZANALYSISSTATEMODIFICATIONDATE + 978307200, 'UNIXEPOCH') AS 'Analysis
 DateTime(ZGENERICASSET.ZADDEDDATE + 978307200, 'UNIXEPOCH') AS 'AddDate',
 ZGENERICASSET.ZFILENAME AS 'FileName',
```

	GenericAsset_zpk	AddAttributes_Key	DetectedFaceAsset	Kind	DateCreated	EXIFtimestamp	SceneAnalysisTim
45	33	33	NULL	Photo	2020-04-11 17:56:09	2020:04:11 13:56:09	2020-04-11 17:56:
46	34	34	NULL	Photo	2020-04-11 17:56:55	2020:04:11 13:56:56	2020-04-11 17:56:
47	35	35	NULL	Photo	2020-04-12 00:45:00	NULL	2020-04-12 00:45
48	36	36	NULL	Photo	2020-04-12 01:29:12	NULL	2020-04-12 01:29:
49	37	37	NULL	Video	2020-04-12 01:57:09	2020:04:11 21:57:09	2020-04-12 01:57:
50	38	38	38	Photo	2020-04-12 12:58:25	NULL	2020-04-12 12:58:
51	39	39	NULL	Photo	2020-04-12 13:25:00	2020:04:12 09:25:00	2020-04-12 13:25:
52	40	40	NULL	Photo	2020-04-12 13:25:26	2020:04:12 09:25:26	2020-04-12 13:25:
53	41	41	NULL	Photo	2020-04-12 13:26:13	2020:04:12 09:26:13	2020-04-12 13:26:
54	42	42	NULL	Photo	2020-04-12 13:26:30	2020:04:12 09:26:30	2020-04-12 13:26:

Figure 9.3 – The results of the query from the Photos.sqlite database

	e	AlbumTitle	CreatorBundleID	EditorBundleID	Directory	UniformI
22	720D5959.jpg	Imgur	NULL	NULL	DCIM/100APPLE	public.jpeg
23	720D5959.jpg	Imgur	NULL	NULL	DCIM/100APPLE	public.jpeg
24		NULL	NULL	com.apple.ScreenshotServicesService	DCIM/100APPLE	public.jpeg
25	A30D6460.jpg	NULL	NULL	NULL	DCIM/100APPLE	public.jpeg
26	0460310D.jpg	NULL	NULL	NULL	DCIM/100APPLE	public.jpeg
27	-13D1A552AFDF.jpg	NULL	NULL	NULL	DCIM/100APPLE	public.jpeg
28		NULL	com.reddit.Reddit	NULL	DCIM/100APPLE	public.jpeg
29		NULL	org.whispersystems.signal	NULL	DCIM/100APPLE	public.png
30		NULL	NULL	NULL	DCIM/100APPLE	public.jpeg
31		NULL	NULL	com.apple.ScreenshotServicesService	DCIM/100APPLE	public.jpeg

Figure 9.4 – CreatorBundleID and EditorBundleID indicate where the asset originates from

	rashDate	ViewCount	PlayCount	ShareCount	LastSharedDate	FileModificationDate	Has_Adjustments	AdjustmentsTimeS
31		2	0	0	*NULL*	2020-04-06 15:02:38	Yes	2020-03-25 01:53:
32		1	0	0	*NULL*	2020-04-06 15:02:38	No	*NULL*
33		2	0	0	*NULL*	2020-04-19 18:25:57	No	*NULL*
34		1	0	0	*NULL*	2020-04-06 15:02:38	No	*NULL*
35		1	0	1	2020-04-05 19:13:52	2020-04-19 18:25:57	No	*NULL*
36		1	0	1	2020-04-05 19:13:52	2020-04-19 18:25:57	No	*NULL*
37		1	0	1	2020-03-27 01:17:03	2020-04-19 18:25:58	No	*NULL*
38		1	0	1	2020-03-27 01:17:03	2020-04-19 18:25:58	No	*NULL*
39		1	0	1	2020-03-27 01:17:03	2020-04-19 18:25:58	No	*NULL*
40		0	0	0	*NULL*	2020-03-27 00:38:01	No	*NULL*

Figure 9.5 – The ShareCount column indicates whether a media asset was shared with other devices

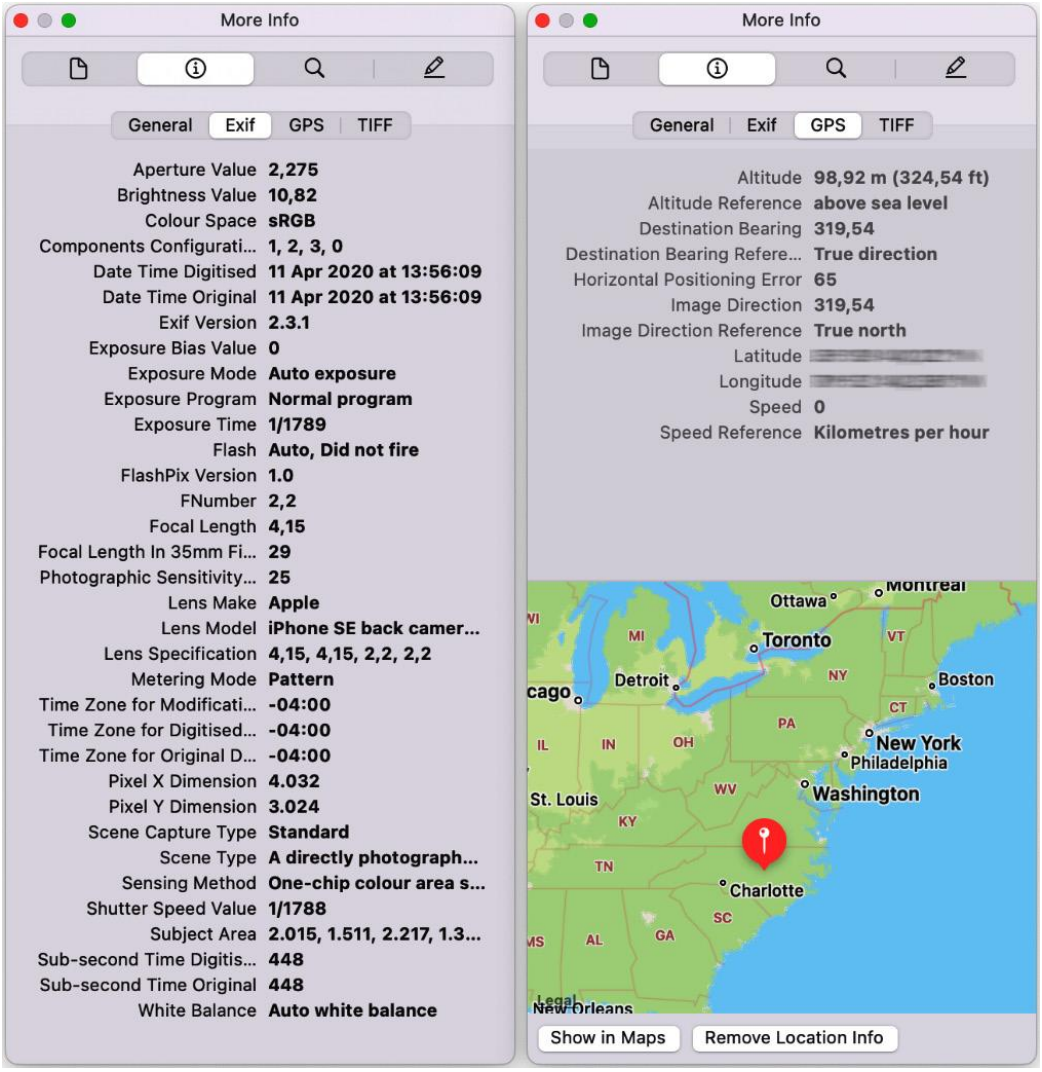


Figure 9.6 – EXIF metadata extracted from a photo that was captured on an iPhone

	ZUUID	ZSTREAMNAME	ZVALUESTRING
	r	/media/nowPlaying	Filter
74	:4698C-793C-46F2-93C0-4C3C5C6A4F2E	/media/nowPlaying	com.apple.podcasts
75	97505-A71C-494E-9CC6-59291287024C	/media/nowPlaying	com.apple.podcasts
76	9B7F4-FEFD-4E8E-931B-359E2EEECC3C	/media/nowPlaying	com.apple.podcasts
77	'22CEE-4834-42B3-9531-E2BDB3F3D1C4	/media/nowPlaying	com.apple.podcasts
78	5BFD9-C420-43F4-B6F2-DB1ADA37DB44	/media/nowPlaying	com.apple.podcasts
79	F2219-190C-4AAF-86C4-5CE72450F13C	/media/nowPlaying	com.apple.podcasts
80	8C843-9905-41D2-9E91-86F9EE1F0399	/media/nowPlaying	com.apple.podcasts
81	3F5AE-9168-4A64-A425-BD6E9A819D53	/media/nowPlaying	com.apple.podcasts
82	38C00-B90C-4B93-80E3-3206D399F163	/media/nowPlaying	com.apple.podcasts
83	ACB47-489B-4563-A46B-EA117D67F931	/media/nowPlaying	com.apple.podcasts
84	004A8-5501-4C06-94C9-7A7CDEAD6708	/media/nowPlaying	com.apple.podcasts
85	014E4-3F63-4A0B-9F98-1DF287575D2B	/media/nowPlaying	com.apple.podcasts
86	E3A15-CC82-4BBD-896A-AEA0C9C93BBF	/media/nowPlaying	com.apple.podcasts

74 - 86 of 281 Go to: 1

Figure 9.7 – Media events extracted from the KnowledgeC.db database

	ZVALUESTRING	NOW PLAYING ALBUM	NOW PLAYING ARTIST	NOW PLAYING GENRE
87	com.apple.Music	B.o.B Presents: The Adventures of Bobby Ray (Deluxe)	B.o.B	Hip-Hop/Rap
88	com.apple.Music	Mother's Milk	Red Hot Chili Peppers	Alternative
89	com.apple.Music	Faster Kill Pussycat (feat.Brittany Murphy)	Oakenfold	Dance
90	com.apple.Music	Mayday	Boys Noize	Breakbeat
91	com.apple.Music	Until Now	Swedish House Mafia	Dance
92	com.apple.Music	Until Now	Swedish House Mafia	Dance
93	com.apple.Music	Until Now	Swedish House Mafia	Dance
94	com.apple.Music	Until Now	Swedish House Mafia	Dance
95	com.apple.Music	Until Now	Swedish House Mafia	Dance
96	com.apple.Music	Until Now	Swedish House Mafia	Dance

Figure 9.8 – Extracting media viewing data from KnowledgeC.db

Code and Commands

Code 9.1:

```
SELECT

Z_PK,

DATETIME(ZDATECREATED + 978307200, 'UNIXEPOCH') AS "Created
date",
```

```
DATETIME(ZMODIFICATIONDATE + 978307200, 'UNIXEPOCH') AS
"Modified date",

DATETIME(ZTRASHEDDATE + 978307200, 'UNIXEPOCH') AS "Deleted
date",

ZFILENAME AS "File name",

ZDIRECTORY AS "Directory",

ZWIDTH AS "Width",

ZHEIGHT AS "Height",

ZHIDDEN AS "Hidden"

FROM ZGENERICASSET

ORDER BY Z_PK ASC;
```

Command 9.2:

```
SELECT

datetime(ZOBJECT.ZSTARTDATE+978307200,'UNIXEPOCH',
'LOCALTIME') as "START",

datetime(ZOBJECT.ZENDDATE+978307200,'UNIXEPOCH',
'LOCALTIME') as "END",

(ZOBJECT.ZENDDATE-ZOBJECT.ZSTARTDATE) as "USAGE IN
SECONDS",

ZOBJECT.ZVALUESTRING,

ZSTRUCTUREDMETADATA.Z_DKNOWPLAYINGMETADATAKEY__ALBUM as
"NOW PLAYING ALBUM",

ZSTRUCTUREDMETADATA.Z_DKNOWPLAYINGMETADATAKEY__ARTIST as
"NOW PLAYING ARTIST",

ZSTRUCTUREDMETADATA.Z_DKNOWPLAYINGMETADATAKEY__GENRE as
"NOW PLAYING GENRE",

ZSTRUCTUREDMETADATA.Z_DKNOWPLAYINGMETADATAKEY__TITLE as
"NOW PLAYING TITLE",
```

```
ZSTRUCTUREDMETADATA.Z_DKNOWPLAYINGMETADATAKEY__DURATION as
"NOW PLAYING DURATION"

FROM ZOBJECT

left join ZSTRUCTUREDMETADATA on
ZOBJECT.ZSTRUCTUREDMETADATA = ZSTRUCTUREDMETADATA.Z_PK

left join ZSOURCE on ZOBJECT.ZSOURCE = ZSOURCE.Z_PK

WHERE ZSTREAMNAME like "/media%"

ORDER BY "START";
```

Links

- *Using Photos.sqlite to Show the Relationships Between Photos and the Application they were Created with?*, *DFIR Review*, *Koenig, S.*, *2021*: https://dfir.pubpub.org/pub/v19rksyf

Chapter 10

Images

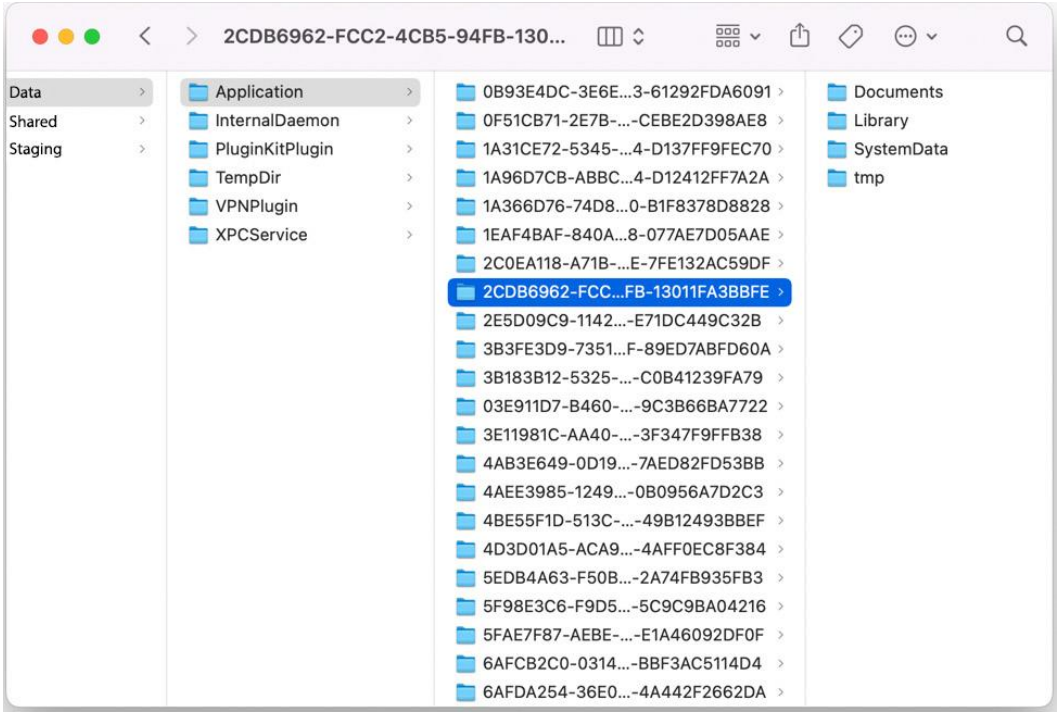


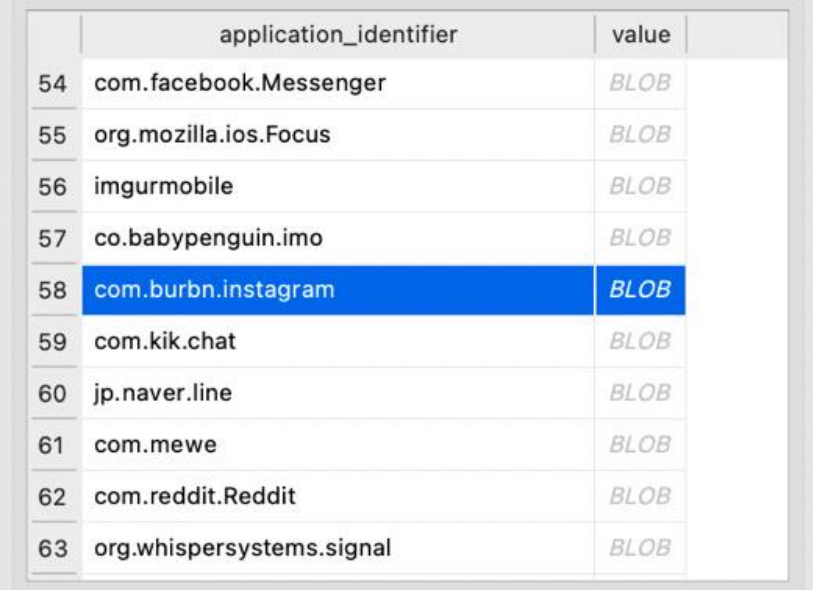
Figure 10.1 – The data container's directory structure

	id	application_identifier
	Filt...	Filter
1	1	com.apple.MobileSMS
2	2	com.apple.Diagnostics
3	3	com.apple.Health
4	4	com.apple.datadetectors.DDActionsService
5	5	com.apple.TVRemoteUIService
6	6	com.apple.Preferences
7	7	com.apple.CarPlaySettings
8	8	com.apple.Home.HomeUIService
9	9	com.apple.Magnifier
10	10	com.apple.BarcodeScanner
11	11	com.apple.Passbook
12	12	com.apple.siri

Figure 10.2 – The application_identifier table from applicationState.db

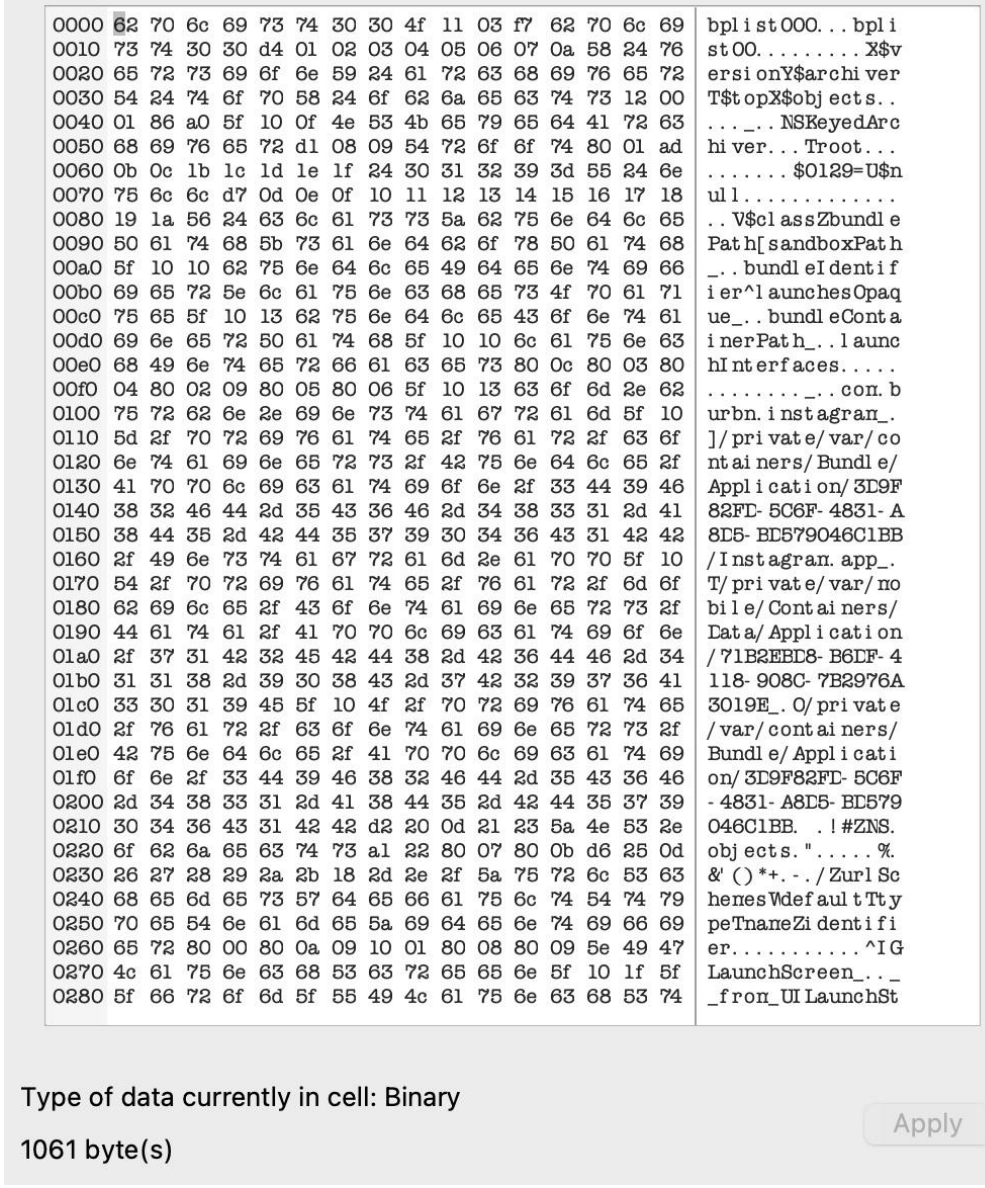
	application_identifier	value
1	com.apple.MobileSMS	BLOB
2	com.apple.Diagnostics	BLOB
3	com.apple.Health	BLOB
4	com.apple.datadetectors.DDActionsService	BLOB
5	com.apple.TVRemoteUIService	BLOB
6	com.apple.Preferences	BLOB
7	com.apple.CarPlaySettings	BLOB
8	com.apple.Home.HomeUIService	BLOB
9	com.apple.Magnifier	BLOB
10	com.apple.BarcodeScanner	BLOB

Figure 10.3 – The query prints bundle IDs for all the installed applications



	application_identifier	value
54	com.facebook.Messenger	BLOB
55	org.mozilla.ios.Focus	BLOB
56	imgurmobile	BLOB
57	co.babypenguin.imo	BLOB
58	com.burbn.instagram	BLOB
59	com.kik.chat	BLOB
60	jp.naver.line	BLOB
61	com.mewe	BLOB
62	com.reddit.Reddit	BLOB
63	org.whispersystems.signal	BLOB

Figure 10.4 – Each Bundle ID has an associated binary BLOB



```
0000 62 70 6c 69 73 74 30 30 4f 11 03 f7 62 70 6c 69  bplist00O...bpli
0010 73 74 30 30 d4 01 02 03 04 05 06 07 0a 58 24 76  st00.........X$v
0020 65 72 73 69 6f 6e 59 24 61 72 63 68 69 76 65 72  ersionY$archiver
0030 54 24 74 6f 70 58 24 6f 62 6a 65 63 74 73 12 00  T$topX$objects..
0040 01 86 a0 5f 10 0f 4e 53 4b 65 79 65 64 41 72 63  ..._..NSKeyedArc
0050 68 69 76 65 72 d1 08 09 54 72 6f 6f 74 80 01 ad  hiver...Troot...
0060 0b 0c 1b 1c 1d 1e 1f 24 30 31 32 39 3d 55 24 6e  .......$0129=U$n
0070 75 6c 6c d7 0d 0e 0f 10 11 12 13 14 15 16 17 18  ull.............
0080 19 1a 56 24 63 6c 61 73 73 5a 62 75 6e 64 6c 65  ..V$classZbundle
0090 50 61 74 68 5b 73 61 6e 64 62 6f 78 50 61 74 68  Path[sandboxPath
00a0 5f 10 10 62 75 6e 64 6c 65 49 64 65 6e 74 69 66  _..bundleIdentif
00b0 69 65 72 5e 6c 61 75 6e 63 68 65 73 4f 70 61 71  ier^launchesOpaq
00c0 75 65 5f 10 13 62 75 6e 64 6c 65 43 6f 6e 74 61  ue_..bundleConta
00d0 69 6e 65 72 50 61 74 68 5f 10 10 6c 61 75 6e 63  inerPath_..launc
00e0 68 49 6e 74 65 72 66 61 63 65 73 80 0c 80 03 80  hInterfaces.....
00f0 04 80 02 09 80 05 80 06 5f 10 13 63 6f 6d 2e 62  ........_..com.b
0100 75 72 62 6e 2e 69 6e 73 74 61 67 72 61 6d 5f 10  urbn.instagram_.
0110 5d 2f 70 72 69 76 61 74 65 2f 76 61 72 2f 63 6f  ]/private/var/co
0120 6e 74 61 69 6e 65 72 73 2f 42 75 6e 64 6c 65 2f  ntainers/Bundle/
0130 41 70 70 6c 69 63 61 74 69 6f 6e 2f 33 44 39 46  Application/3D9F
0140 38 32 46 44 2d 35 43 36 46 2d 34 38 33 31 2d 41  82FD-5C6F-4831-A
0150 38 44 35 2d 42 44 35 37 39 30 34 36 43 31 42 42  8D5-BD579046C1BB
0160 2f 49 6e 73 74 61 67 72 61 6d 2e 61 70 70 5f 10  /Instagram.app_.
0170 54 2f 70 72 69 76 61 74 65 2f 76 61 72 2f 6d 6f  T/private/var/mo
0180 62 69 6c 65 2f 43 6f 6e 74 61 69 6e 65 72 73 2f  bile/Containers/
0190 44 61 74 61 2f 41 70 70 6c 69 63 61 74 69 6f 6e  Data/Application
01a0 2f 37 31 42 32 45 42 44 38 2d 42 36 44 46 2d 34  /71B2EBD8-B6DF-4
01b0 31 31 38 2d 39 30 38 43 2d 37 42 32 39 37 36 41  118-908C-7B2976A
01c0 33 30 31 39 45 5f 10 4f 2f 70 72 69 76 61 74 65  3019E_.O/private
01d0 2f 76 61 72 2f 63 6f 6e 74 61 69 6e 65 72 73 2f  /var/containers/
01e0 42 75 6e 64 6c 65 2f 41 70 70 6c 69 63 61 74 69  Bundle/Applicati
01f0 6f 6e 2f 33 44 39 46 38 32 46 44 2d 35 43 36 46  on/3D9F82FD-5C6F
0200 2d 34 38 33 31 2d 41 38 44 35 2d 42 44 35 37 39  -4831-A8D5-BD579
0210 30 34 36 43 31 42 42 d2 20 0d 21 23 5a 4e 53 2e  046C1BB. .!#ZNS.
0220 6f 62 6a 65 63 74 73 a1 22 80 07 80 0b d6 25 0d  objects."....%.
0230 26 27 28 29 2a 2b 18 2d 2e 2f 5a 75 72 6c 53 63  &'()*+.-./ZurlSc
0240 68 65 6d 65 73 57 64 65 66 61 75 6c 74 54 74 79  hemesWdefaultTty
0250 70 65 54 6e 61 6d 65 5a 69 64 65 6e 74 69 66 69  peTnameZidentifi
0260 65 72 80 00 80 0a 09 10 01 80 08 80 09 5e 49 47  er...........^IG
0270 4c 61 75 6e 63 68 53 63 72 65 65 6e 5f 10 1f 5f  LaunchScreen_.._
0280 5f 66 72 6f 6d 5f 55 49 4c 61 75 6e 63 68 53 74  _from_UILaunchSt
```

Type of data currently in cell: Binary

1061 byte(s)

Apply

Figure 10.5 – The BLOB contains a binary PLIST

instagram.plist — Open with TextEdit

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>$archiver</key>
        <string>NSKeyedArchiver</string>
        <key>$objects</key>
        <array>
                <string>$null</string>
                <dict>
                        <key>$class</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>12</integer>
                        </dict>
                        <key>bundleContainerPath</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>5</integer>
                        </dict>
                        <key>bundleIdentifier</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>2</integer>
                        </dict>
                        <key>bundlePath</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>3</integer>
                        </dict>
                        <key>launchInterfaces</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>6</integer>
                        </dict>
                        <key>launchesOpaque</key>
                        <true/>
                        <key>sandboxPath</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>4</integer>
                        </dict>
                </dict>
                <string>com.burbn.instagram</string>
                <string>/private/var/containers/Bundle/Application/3D9F82FD-5C6F-4831-A8D5-BD579046C1BB/Instagram.app</
string>
                <string>/private/var/mobile/Containers/Data/Application/71B2EBD8-B6DF-4118-908C-7B2976A3019E</string>
                <string>/private/var/containers/Bundle/Application/3D9F82FD-5C6F-4831-A8D5-BD579046C1BB</string>
                <dict>
                        <key>$class</key>
                        <dict>
                                <key>CF$UID</key>
```

Figure 10.6 – The PLIST contains details related to the app's containers

Figure 10.7 – The fsmon tool prints any filesystem events

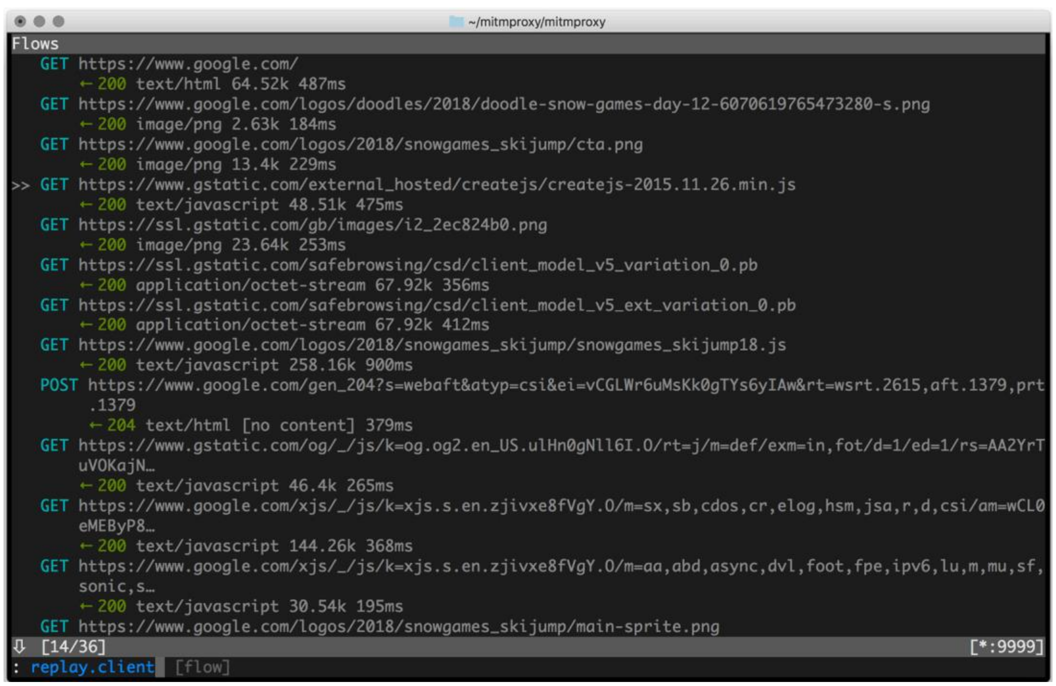


Figure 10.8 – Intercepting network traffic using mitmproxy

Tables

Social networking applications

In this section, we will focus on the three primary social networking applications, **Facebook**, **Instagram**, and **Twitter**. The **Reddit** app is also included in this category.

Facebook

The following table shows the most common artifacts for Facebook:

Artifact	Description
Library/Preferences/com.facebook.Facebook.plist	User preferences and details of logged-in users (email and Facebook ID).
Library/Caches/com.facebook.Facebook.*/	Cached files and images, such as media, are stored while browsing through posts and pages.

Table 10.1 – Common Facebook artifacts

Instagram

The following table shows the most common artifacts for Instagram:

Artifact	Description
Library/Preferences/com.burbn.instagram.plist	User preferences and details of logged-in users (email and Instagram ID)
Library/Application Support/ DirectSQLiteDatabase/*.db	SQLite database containing user data
Library/Application Support/<User_ID>/ pending-requests.plist	Property List containing pending friend and message requests
Library/Caches/com.burbn.instagram. IGImageCache/*	Cached images
Library/Caches/com.burbn.instagram. IGSparseVideoCache/*	Cached videos

Table 10.2 – Common Instagram artifacts

Twitter

The following table shows the most common artifacts for Twitter:

Artifact	Description
Library/Preferences/com.atebits.Tweetie2.plist	User preferences and details of logged-in users
Library/Caches/com.twitter.api.cache/Cache.db	SQLite database containing cached API requests, such as visited profiles
Library/SplashBoard/Snapshots/*	App-generated screenshots
Library/Caches/TIPImagePipeline/*	Cached images
Library/Caches/com.atebits.Tweetie2/fsCachedData/*	Cached data

Table 10.3 – Common Twitter artifacts

Reddit

The following table shows the most common artifacts for Reddit:

Artifact	Description
Library/Preferences/com.reddit.Reddit.plist	User preferences and details of logged-in users
Documents/release2/accountData/*	Binary Property Lists containing user data
Library/SplashBoard/Snapshots/*	App-generated screenshots
Library/Caches/com.reddit.Reddit/Cache.db	SQLite database containing cached API requests
Library/Caches/com.reddit.Reddit/imagedownload/*	SQLite database containing cached images

Table 10.4 – Common Reddit artifacts

Now, let's look at a few messaging apps.

Messaging applications

Although messaging applications were discussed in detail in *Chapter 8*, *Email and Messaging Forensics*, this section will provide a reference for some of the most commonly used instant messaging apps, such as **WhatsApp**, **Telegram**, **Facebook Messenger**, and **Signal**.

WhatsApp

The following table shows the most common artifacts for WhatsApp:

Artifact	Description
Documents/CallHistory.sqlite	Audio and video call history
Documents/blockedcontacts.dat	Contacts blocked by user
Library/2fa.plist	Propery List indicating if multi-factor authentication is enabled
Library/Caches/*	Cached media and API requests
Library/Preferences/net.whatsapp.WhatsApp.plist	User preferences
Library/SplashBoard/Snapshots/*	App-generated screenshots
Shared/AppGroup/*/ChatStorage.sqlite	SQLite database storing messages
Shared/AppGroup/*/ContactsV2.sqlite	SQLite database storing contact details
Shared/AppGroup/*/Library/Preferences/*	Property Lists storing preferences

Table 10.5 – Common WhatsApp artifacts

Telegram

The following table shows the most common artifacts for Telegram:

Artifact	Description
Library/Preferences/ph.telegra.Telegraph.plist	Property List containing user preferences
Library/Caches/ph.telegra.Telegraph/fsCachedData/*	Cached media and API requests
Shared/AppGroup/telegram-data/account-<Account_ID>/postbox/db/db_sqlite	SQLite database containing chats and messages
Shared/AppGroup/telegram-data/account-<Account_ID>/postbox/media/*	Chat media files

Table 10.6 – Common Telegram artifacts

Facebook Messenger

The following table shows the most common artifacts for Facebook Messenger:

Artifact	Description
Documents/com.facebook.Messenger.preferences/*	Property List containing user preferences
Library/Preferences/com.facebook.Messenger.plist	Application settings
Shared/AppGroup/shared_messenger_contacts/*.db	SQLite database containing contacts
Shared/AppGroup/shared_messenger_messages/*.db	SQLite database that stores chats and messages
Shared/AppGroup/lightspeed-imageCache/*	Cache

Table 10.7 – Common Facebook Messenger artifacts

Signal

The following table shows the most common artifacts for Signal:

Artifact	Description
Library/Preferences/org.whispersystems.signal.plist	Application settings
Library/Caches/*	Cached media and API requests
Shared/AppGroup/Attachments/*	Media files
Shared/AppGroup/grdb/signal.sqlite	SQLite database containing chats and messages

Table 10.8 – Common Signal artifacts

Next, we'll discuss artifacts related to productivity applications.

Productivity applications

The following tables describe artifacts concerning work and productivity applications, such as **Microsoft Teams**, **Zoom**, **Dropbox**, **Microsoft OneDrive**, and **Gmail**.

Microsoft Teams

The following table shows the most common artifacts for Microsoft Teams:

Artifact	Description
Library/Preferences/com.microsoft.skype.teams.plist	User preferences and application settings
SplashBoard/Snapshots/*	App-generated screenshots
Shared/AppGroup/Library/Preferences/*	Property Lists containing user data

Table 10.9 – Common Microsoft Teams artifacts

Zoom

The following table shows the most common artifacts for Zoom:

Artifact	Description
Library/Preferences/us.zoom.videomeetings.plist	User preferences and application settings
Documents/Data/zoommeeting.db	SQLite database containing data related to Zoom meetings and messages exchanged during a meeting
Documents/Data/zoomus.db	SQLite database that stores user data

Table 10.10 – Common Zoom Artifacts

Dropbox

The following table shows the most common artifacts for Dropbox:

Artifact	Description
Library/Preferences/com.getdropbox.Dropbox.plist	User preferences and application settings
Documents/Users/<id>/Dropbox.sqlite	SQLite database containing user data and file information
Documents/Users/<id>/metadata.sqlite	SQLite database that stores the metadata of files stored on Dropbox

Table 10.11 – Common Dropbox artifacts

Microsoft OneDrive

The following table shows the most common artifacts for Microsoft OneDrive:

Artifact	Description
Library/Preferences/com.microsoft.skydrive.plist	User preferences and application settings
Library/Database/moddatabase.db	SQLite database containing user data
Shared/AppGroup/OneDrive/StramCacheQT/*	Cached files

Table 10.12 – Common OneDrive artifacts

Gmail

The following table shows the most common artifacts for Gmail:

Artifact	Description
Library/Preferences/com.google.Gmail.plist	User preferences, application settings, and logged-in user data
Library/ApplicationSupport/data/<email>/*	SQLite database containing user data
Library/Caches/*	Cached files

Table 10.13 – Common Gmail artifacts

Next, we'll look at a few multimedia apps.

Multimedia applications

This section includes applications that allow you to view or share media files, such as **Netflix**, **YouTube**, **Spotify**, **Snapchat**, and **TikTok**.

Netflix

The following table shows the most common artifacts for Netflix:

Artifact	Description
Library/Preferences/com.netflix.Netflix.plist	User preferences and application settings
Documents/store.sqlite	SQLite database containing information on the logged-in user
Library/Caches/*	Cached video streams and API requests

Table 10.14 – Common Netflix artifacts

YouTube

The following table shows the most common artifacts for YouTube:

Artifact	Description
Library/Preferences/com.google.ios.youtube.plist	User preferences and application settings
Library/Caches/*	Video and API requests caches

Table 10.15 – Common YouTube artifacts

Spotify

The following table shows the most common artifacts for Spotify:

Artifact	Description
Library/Preferences/com.spotify.client.plist	User preferences and application settings
Documents/InstrumentsModel.sqlite	SQLite database that stores app events
Library/ApplicationSupport/PersistentCache/mercury.db	Playback history cached media
Library/ApplicationSupport/Users/<id>/recently_played	Recently played history in binary format
SplashBoard/Snapshots/*	App-generated screenshots

Table 10.16 – Common Spotify artifacts

Snapchat

The following table shows the most common artifacts for Snapchat:

Artifact	Description
Library/Preferences/com.toyopagroup.picaboo.plist	User preferences and application settings
Documents/user.plist	Property List that stores details regarding the logged-in user
Documents/chatConversationStore.plist	Property List containing chats
Documents/stories.plist	Property List that stores story data
Library/Caches/*	Cached media and API requests

Table 10.17 – Common Snapchat artifacts

TikTok

The following table shows the most common artifacts for TikTok:

Artifact	Description
Library/Preferences/com.zhiliaoapp.musically.plist	User preferences and application settings
Documents/AwemeIM.db	SQLite database that stores TikTok contacts
Documents/Aweme.db	SQLite database that contains published TikTok videos
Library/ApplicationSupport/ChatFiles/<id>/db.sqlite	SQLite database that stores chats and messages
Library/Caches/*	Cached media and API requests

Table 10.18 – Common TikTok artifacts

Code and Commands

Command 10.1:

```
SELECT

application underscore identifier underscore
tab.[application underscore identifier],

kvs.[value]
```

```
FROM kvs, key underscore tab,application underscore
identifier underscore tab

WHERE key underscore tab.[id] = '1'

AND kvs.[key] = key underscore tab.[id]

AND application underscore identifier underscore tab.[id] =
kvs.[application underscore identifier]

ORDER BY application underscore identifier underscore
tab.[id];
```

Command 10.2:

```
brew install libimobiledevice
```

Command 10.3:

```
iproxy 4242 44
```

Command 10.4:

```
ssh root@127.0.0.1 -p 4242
```

Command 10.5:

```
scp -P 4242 cda root@127.0.0.1:/usr/bin/
```

Command 10.6:

```
ls -la /usr/bin/
```

Command 10.7:

```
iPhone:~ root# cda instagram

[1] Instagram (com.burbn.instagram)

Bundle:
/private/var/containers/Bundle/Application/D054F8DC-95A4-
4489-BBBD-68F3E457A575

Data:
/private/var/mobile/Containers/Data/Application/2A2FEE52-
B59D-42F1-A810-333364E12525
```

Command 10.8:

```
scp -P 4242 fsmon root@127.0.0.1:/usr/bin/
```

Links

- Download and install the `libimobiledevice` library from https://libimobiledevice.org
- The CDA tool can be downloaded from https://github.com/ay-kay/cda
- 64-bit ARM Binaries for CDA https://github.com/tiepologian/iOS-Tools
- `Fsmon` can be downloaded from GitHub at https://github.com/nowsecure/fsmon. Pre-built binaries for 64-bit ARM devices can also be downloaded from https://github.com/tiepologian/iOS-Tools.
- To install `mitmproxy`, go to https://mitmproxy.org and download the installer. If your workstation has `brew` installed, you can install the proxy by running the following command from your terminal:

  ```
  brew install mitmproxy
  ```

- A complete tour of all Frida's features is beyond the scope of this book; you can get started with the tool by following the tutorials at https://frida.re/docs/home or by reading Alexander Fadeev's blog articles; for example, https://fadeevab.com/quick-start-with-frida-to-reverse-engineer-any-ios-application/.
- To learn about Meduza and download the tool, visit the project's page at https://opensourcelibs.com/lib/meduza.

Chapter 11

Images

Figure 11.1 – Elcomsoft iOS Forensic Toolkit can acquire locked devices using pairing records

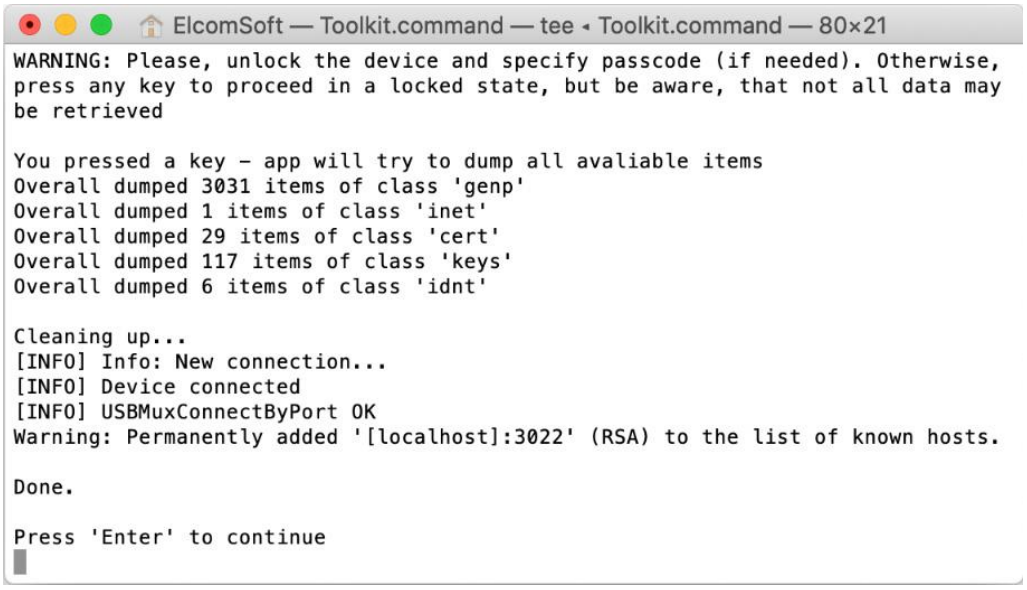


Figure 11.2 – Elcomsoft iOS Forensic Toolkit performing a BFU acquisition

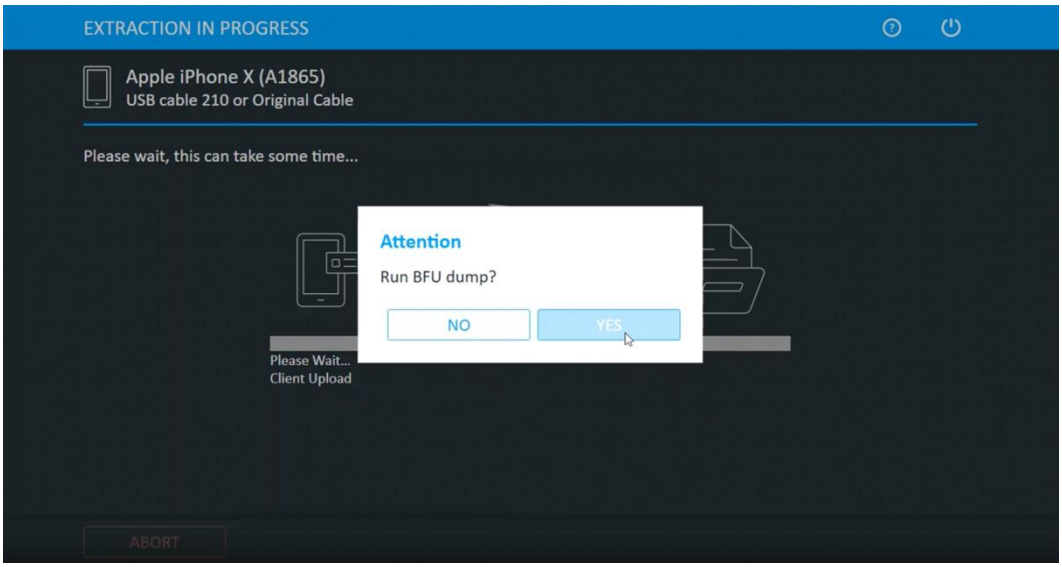


Figure 11.3 – Cellebrite UFED performing a BFU acquisition

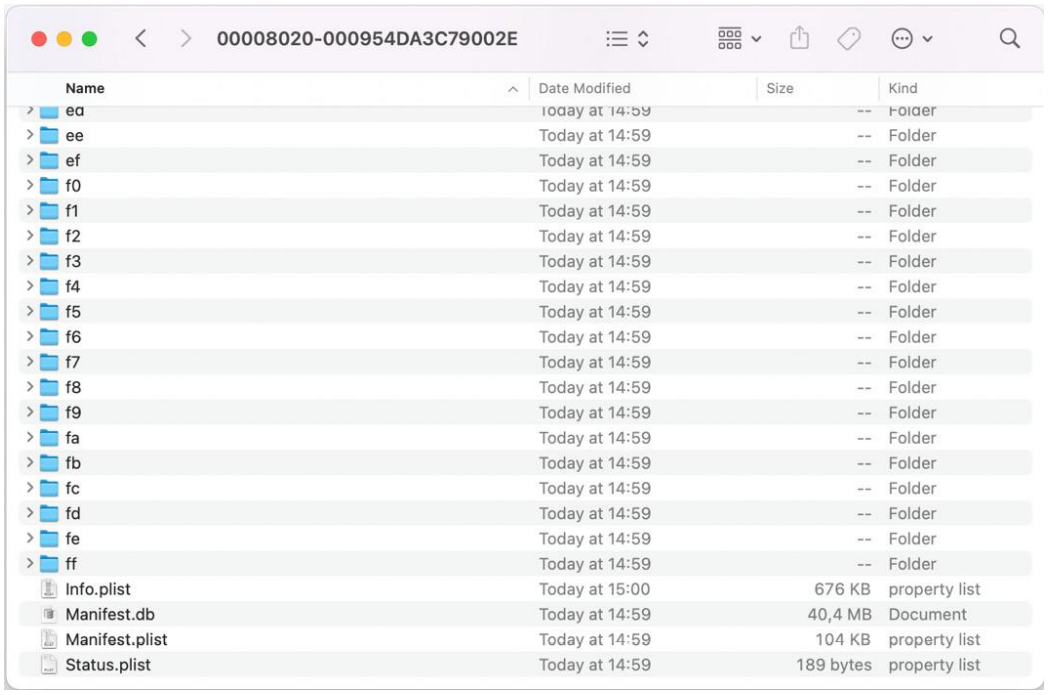


Figure 11.4 – File structure of an iTunes backup

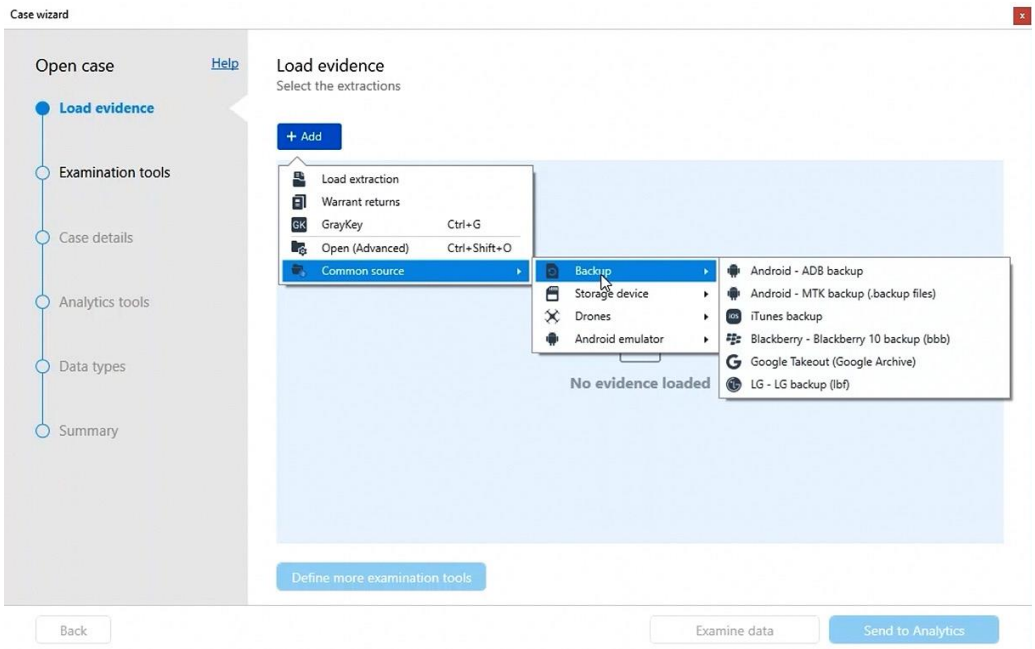


Figure 11.5 – Loading an iTunes backup into Cellebrite Physical Analyzer

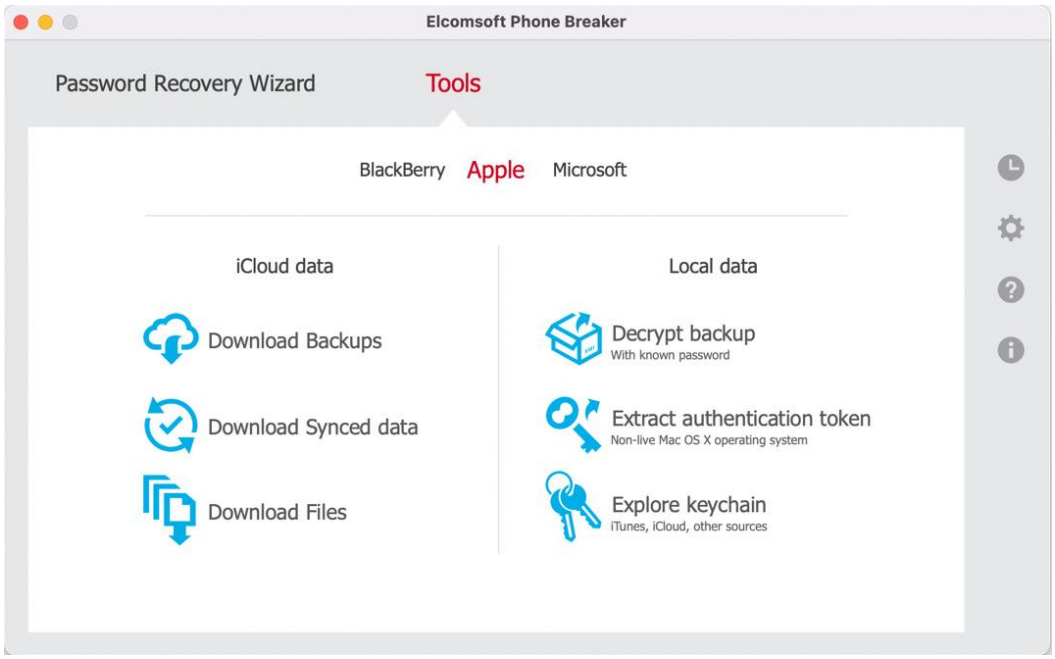


Figure 11.6 – Elcomsoft Phone Breaker home screen

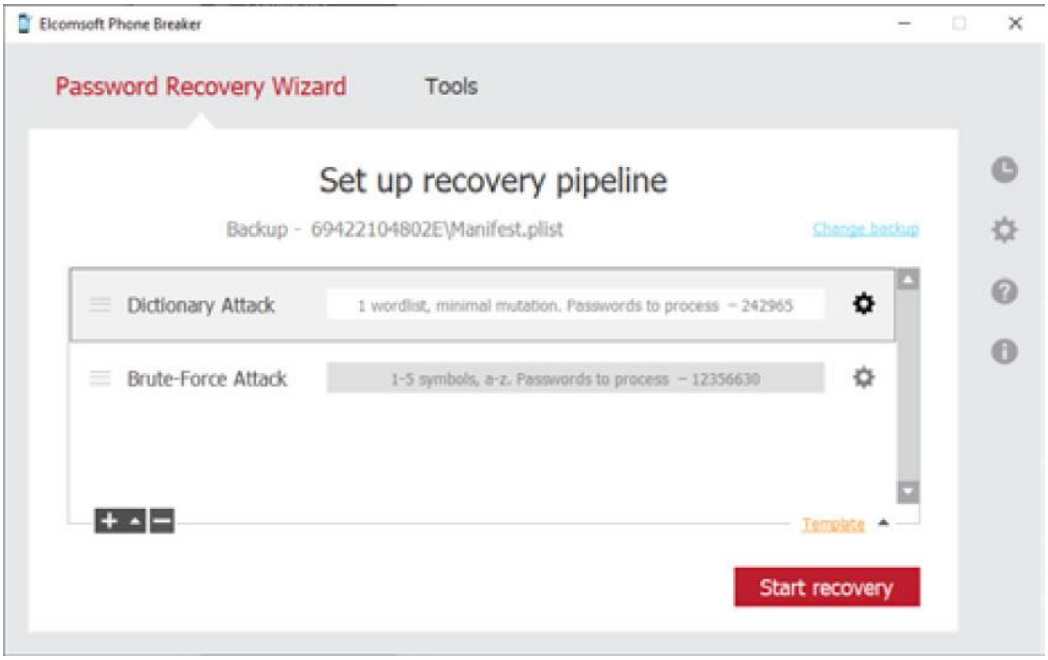


Figure 11.7 – Set up a dictionary attack or a brute-force attack

Figure 11.8 – Apps using iCloud from the device settings

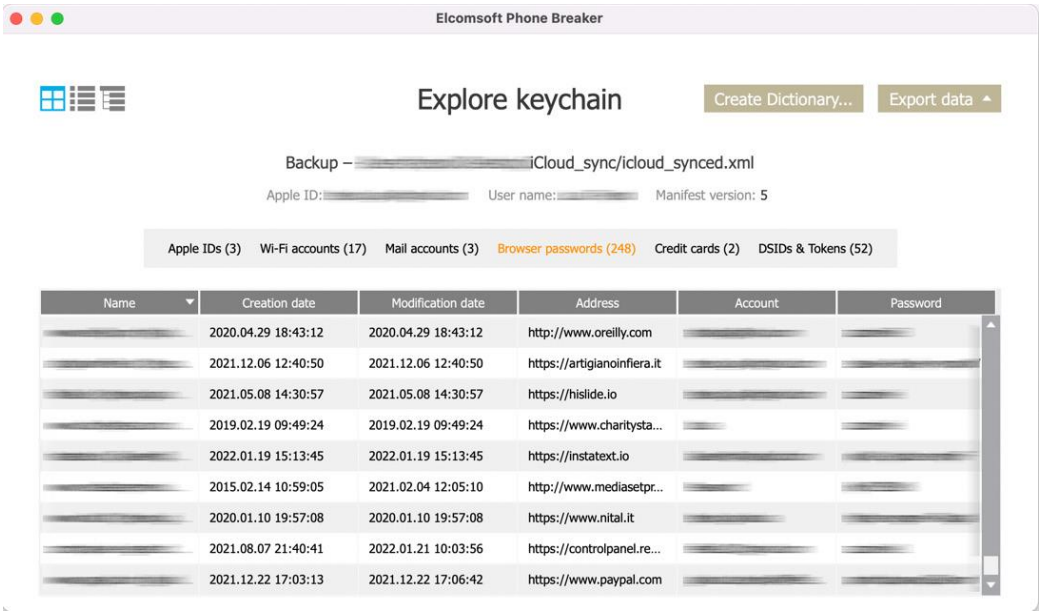


Figure 11.9 – Analyzing iCloud Keychain in Elcomsoft Phone Breaker

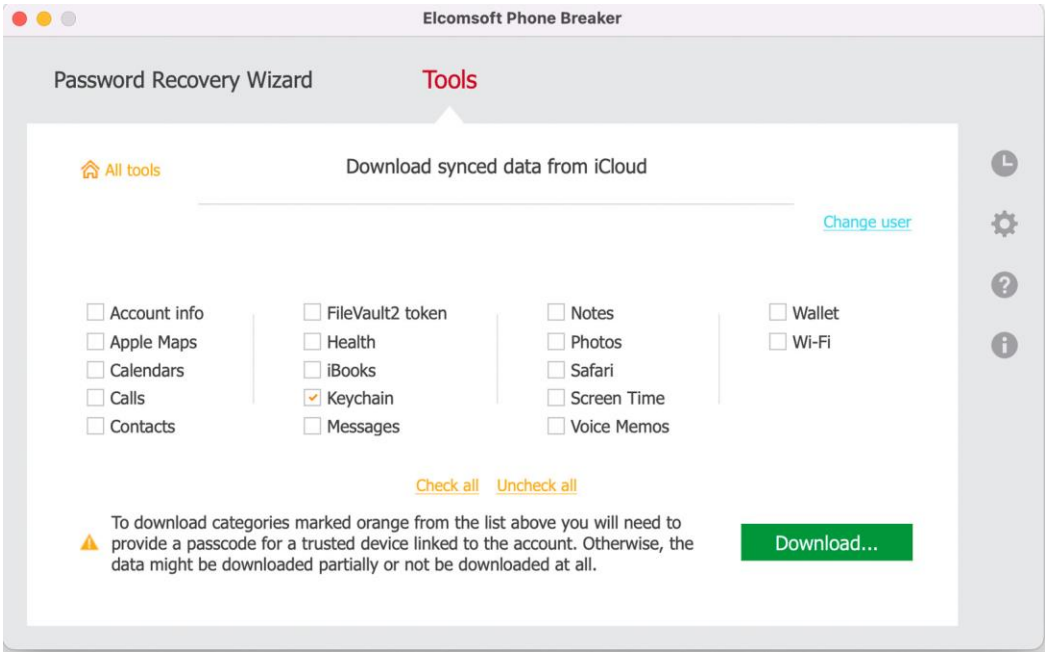


Figure 11.10 – Downloading iCloud synced data with Elcomsoft Phone Breaker

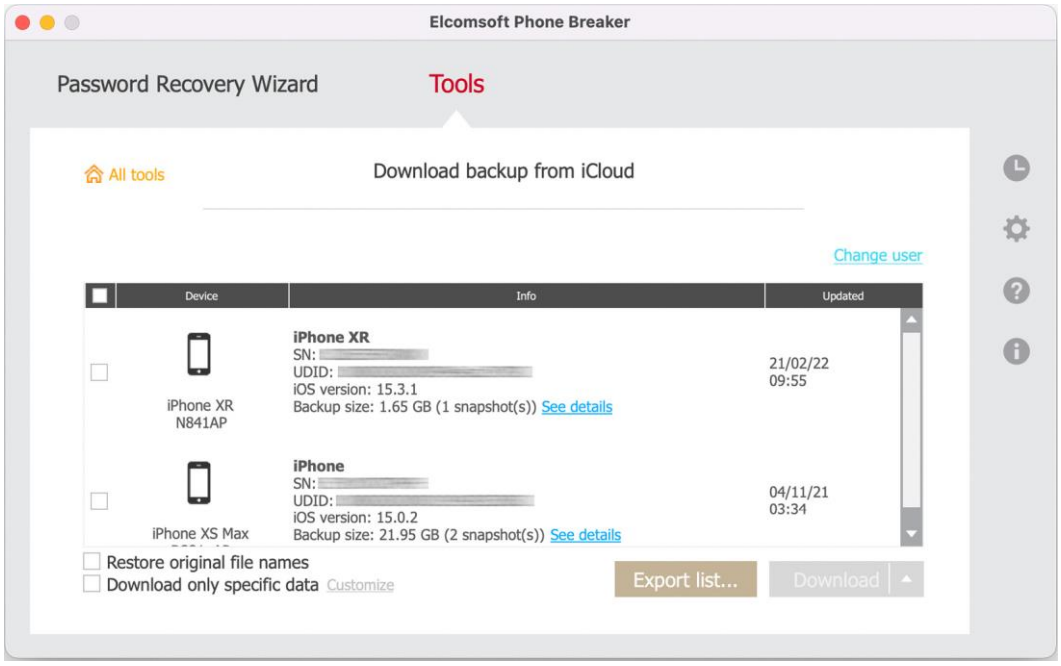


Figure 11.11 – Downloading iCloud backups using Elcomsoft Phone Breaker

Tables

Artifacts
/private/var/mobile/Library/Preferences/*
/private/var/preferences/SystemConfiguration/*
/private/var/root/Library/Preferences/*
/private/var/mobile/Library/Accounts/Accounts3.sqlite
/private/var/mobile/Library/DataAccess/*
/private/installd/Library/Logs/MobileInstallation/*
/private/var/mobile/Library/SpringBoard/*
/private/var/mobile/Library/CallHistoryDB/CallHistoryTemp.storedata
/private/var/mobile/Library/Preferences/com.apple.*
/private/var/mobile/Library/SMS/sms-temp.db
/private/var/mobile/Library/Preferences/com.apple.preferences.network.plist
/private/var/wireless/Library/Databases/CellularUsage.db
/private/var/root/Library/Caches/locationd/consolidated.db

Table 11.1 – Common artifacts found in BFU acquisition

Chapter 12

Images

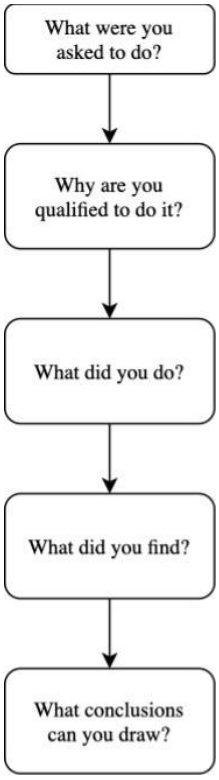


Figure 12.1 – The logical steps of a forensic report

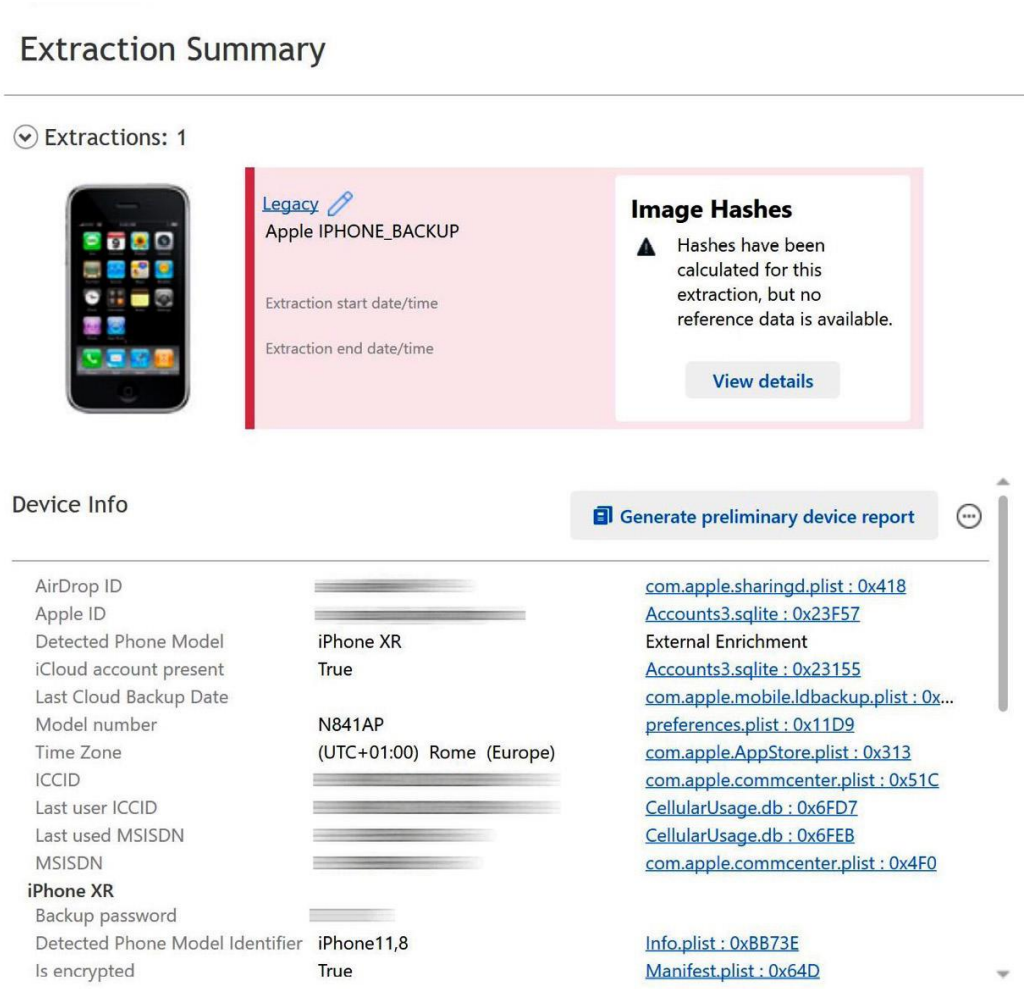


Figure 12.2 – A preliminary device report generated from Extraction Summary

Preliminary Device Report - Apple iOS iTunes (Backup)

Cellebrite
www.cellebrite.com

Device Information

Name	Value
Legacy	
iPhone XR	
MSISDN	
Backup password	
OS Version	15.3.1
Unique ID	00008020-000954DA3C79002E
Detected Phone Model Identifier	iPhone11,8
Serial	
Is encrypted	True
Owner Name	iPhone XR
ICCID	
IMEI	
Phone Settings	
Message Retention Duration	Forever
Location Services Enabled	True
Find my iPhone enabled	True
Apple ID	
iCloud account present	True
Model number	N841AP
Last user ICCID	
Last used MSISDN	
AirDrop ID	
Last Cloud Backup Date	
Time Zone	(UTC+01:00) Rome (Europe)
ICCID	
MSISDN	
Detected Phone Model	iPhone XR
Tethering	
Last Hotspot Activity	2/27/2022 3:04:50 PM(UTC+0)

User Accounts (22)

#	Source	Account Name	User name	Service Type	Creation time	Entries
1				6DE9BF4F-3851-426C-9A1B-EF52BBFE9881	11/6/2021 12:06:09 PM(UTC+0)	**Unknown:** Account Description IMAPMail
2				85BDDDF2-0CBC-4816-A8FB-641B96C0F253	11/6/2021 12:06:13 PM(UTC+0)	**Unknown:** Account Description Device Locator
3				D54A7D3C-C1D3-47B2-8654-54A56472136C	11/6/2021 12:06:14 PM(UTC+0)	**Unknown:** Account Description Find My Friends
4				A78F32F8-204B-488B-B08A-83E1D4CBB24D	11/6/2021 12:06:16 PM(UTC+0)	**Unknown:** Account Description iTunes Store
5				7482D0CF-21B1-4364-A123-59B088C5F3D9	11/6/2021 12:13:33 PM(UTC+0)	**Unknown:** Account Description iTunes Store (Sandbox)
6				A95E99D6-4D90-4B86-95F3-9AEF0AE27DE5	11/6/2021 12:22:03 PM(UTC+0)	**Unknown:** Account Description Gmail
7				91916FC8-758F-493E-8120-FB4C9C22944D	11/8/2021 1:04:22 PM(UTC+0)	**Unknown:** Account Description Gmail
8				03C2C429-2110-4BFA-B3C4-D77117447F29		**Unknown:** Account Description iTunes Store
9				F64E6AC0-E358-443B-B4E4-416C8E348B3C	11/6/2021 12:05:30 PM(UTC+0)	**Unknown:** Account Description IDMS
10				B16C129F-9256-4A70-B52D-DEEFC72B71E1	11/6/2021 12:05:30 PM(UTC+0)	**Unknown:** Account Description Apple ID
11				6E6A2B8C-F745-4E5F-B53D-24F813A35442	11/6/2021 12:05:38 PM(UTC+0)	**Unknown:** Account Description Game Center
12				CB10DC6C-45F7-401E-865C-5725791F5843	11/6/2021 12:05:38 PM(UTC+0)	**Unknown:** Account Description Messages
13				BBD32FAF-A172-4067-A05A-F54E728DE6B6	11/6/2021 12:05:38 PM(UTC+0)	**Unknown:** Account Description CloudKit

Figure 12.3 – An example of a preliminary device report

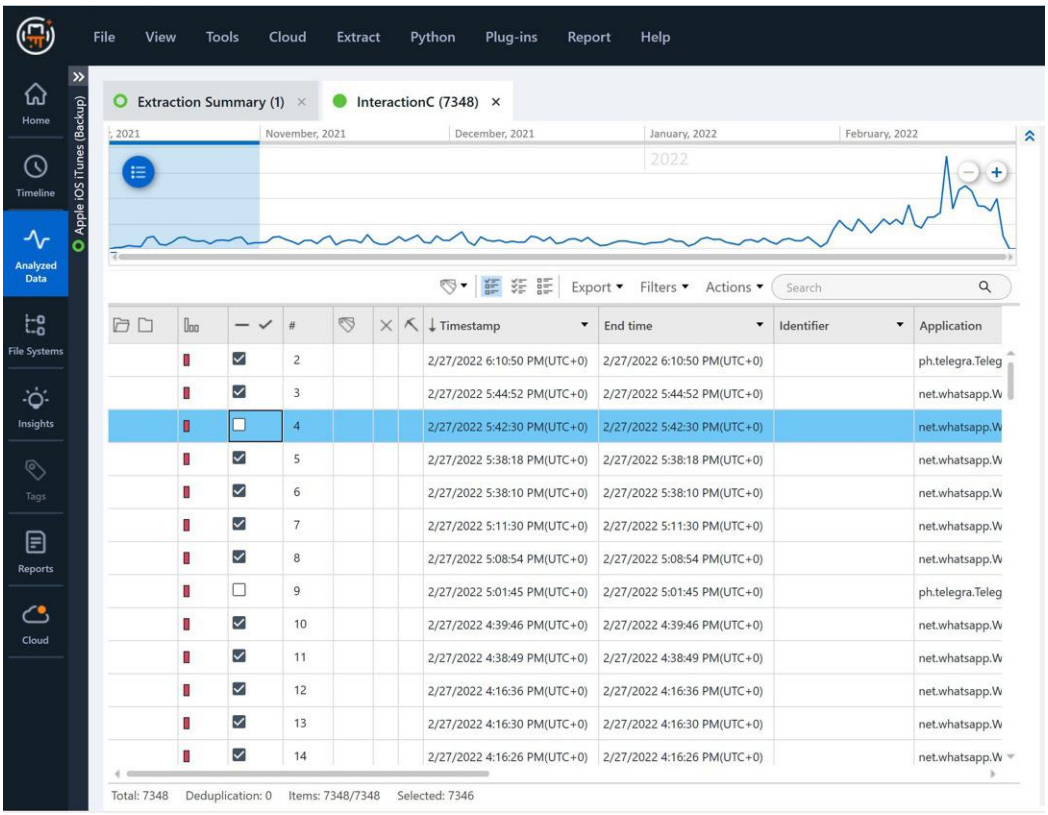


Figure 12.4 – An artifact will be included in the report if the checkbox is enabled

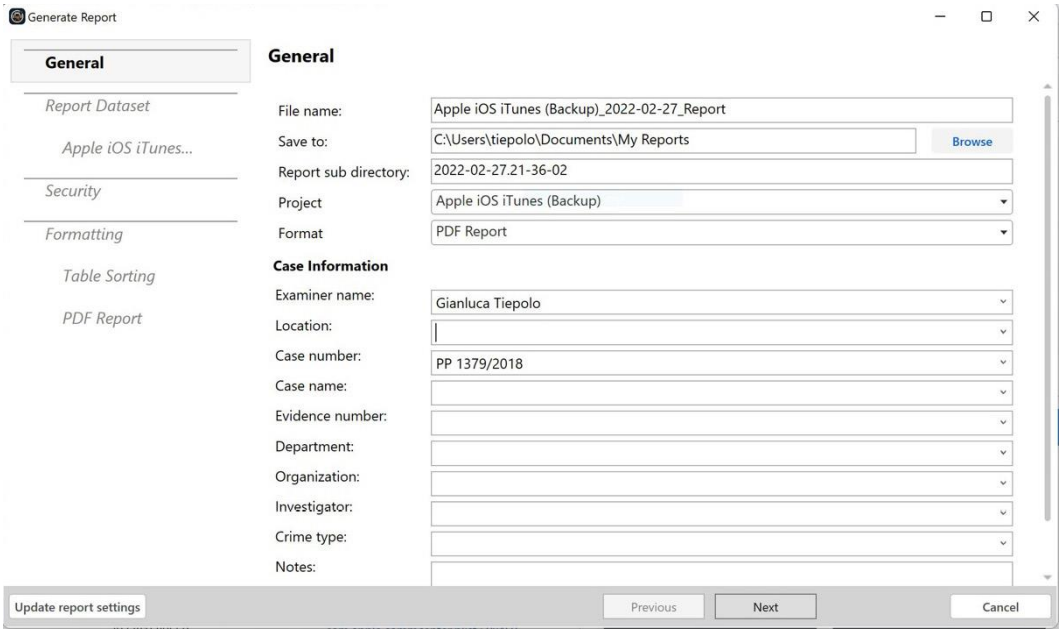


Figure 12.5 – The Generate Report window

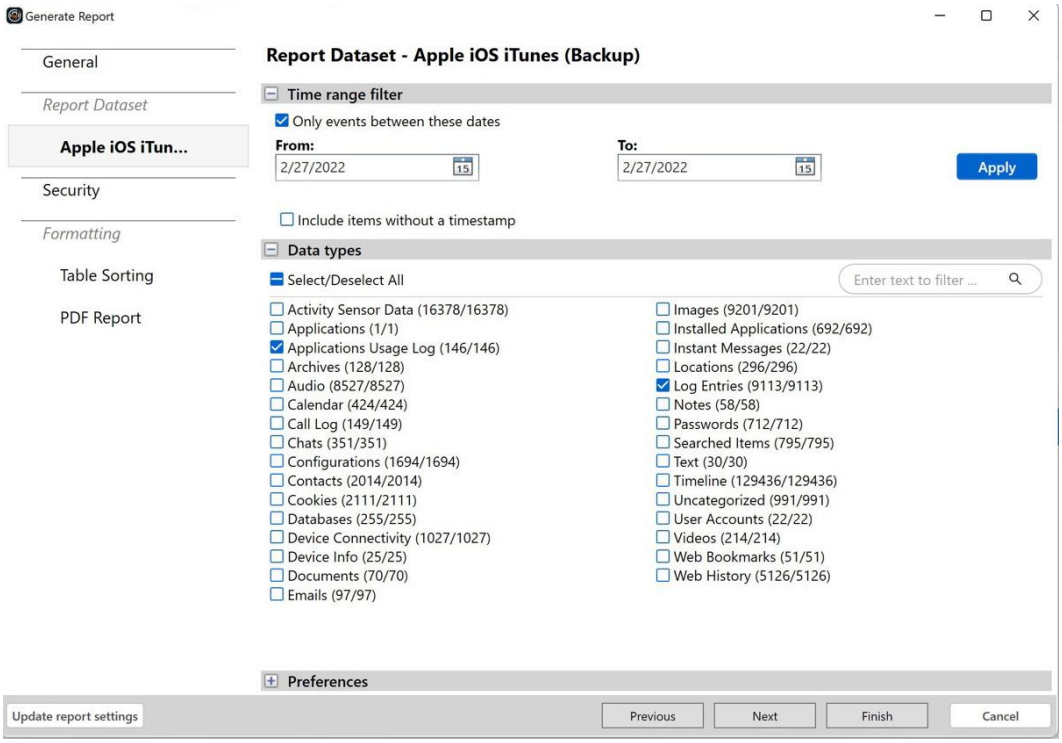


Figure 12.6 – The Report Dataset window

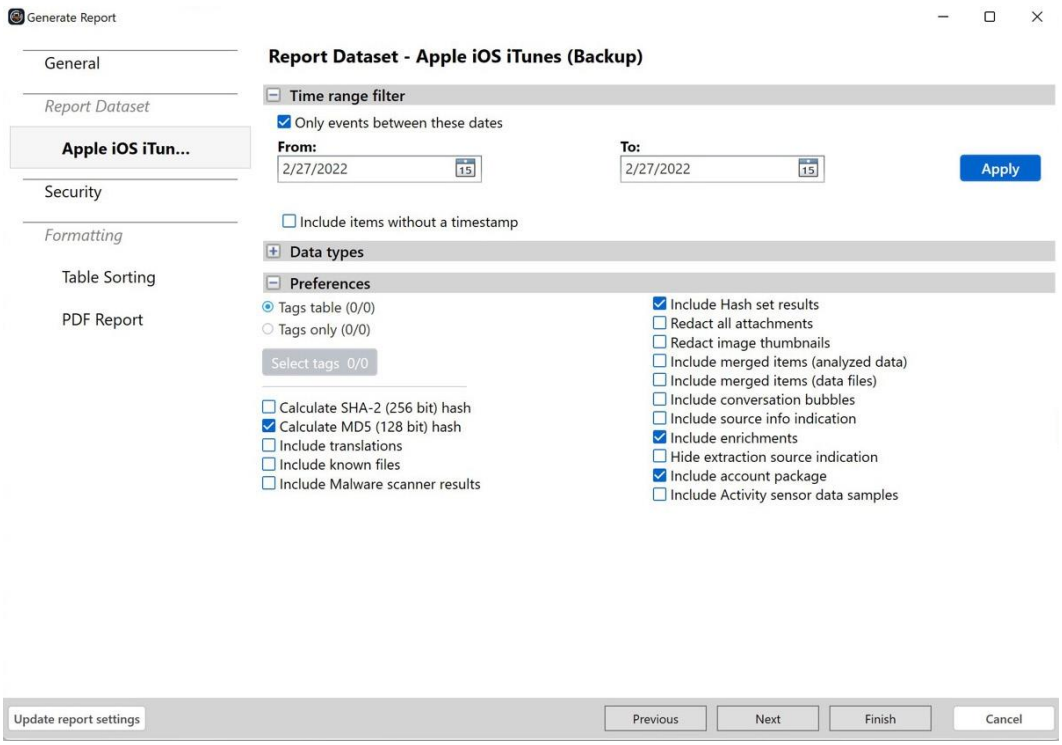
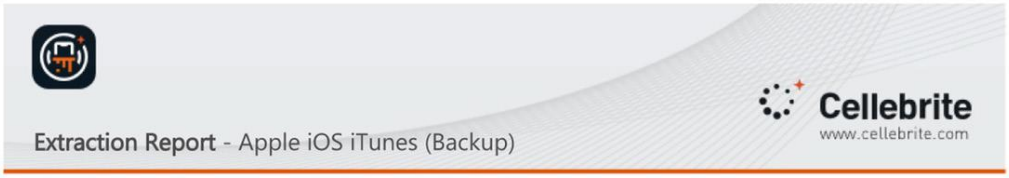


Figure 12.7 – The Preferences tab



Summary

Cellebrite Physical Analyzer version	7.53
Report creation time	2/27/2022 9:39:23 PM +01:00
Time zone settings (UTC)	Original UTC value
Examiner name	Gianluca Tiepolo
Case number	PP 1379/2018

Source Extraction

Legacy	
Selected manufacturer	Apple
Selected device name	IPHONE_BACKUP

Image Hash Details (1)

⚠ Hashes have been calculated for this extraction, but no reference data is available.

#	Name	Info	
1	Folder	Path	00008020-000954DA3C79002E
		Size (bytes)	4024701501
		SHA256	B293C94E273BC3FBDD88E277017805724B6FB480D0BC0AAEF7ED7E575B2A4326

Plugins

#	Name	Author	Version
1	PreProject	Cellebrite	2.0
2	iPhone Backup Parser Parses all iPhone Backup/Logical/FS dumps, including decryption and/or FileSystem creation when necessary	Cellebrite	2.0
3	ContactsCrossReference Cross references the phone numbers in a device's contacts with the numbers in SMS messages and Calls. Will fill in the Name field of calls and SMS if there's a match.	Cellebrite	2.0
4	ProjectProcessorFinisher	Cellebrite	2.0
5	PostProject	Cellebrite	2.0

Figure 12.8 – The Extraction Report Summary

118		ph.telegra.Telegraph		outgoing message	2/26/2022 10:29:50 PM(UTC+0) **End time:** 2/26/2022 10:29:50 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
119		ph.telegra.Telegraph		outgoing message	2/26/2022 10:29:37 PM(UTC+0) **End time:** 2/26/2022 10:29:37 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
120		ph.telegra.Telegraph		incoming message	2/26/2022 10:29:34 PM(UTC+0) **End time:** 2/26/2022 10:29:34 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
121		ph.telegra.Telegraph		incoming message	2/26/2022 10:29:32 PM(UTC+0) **End time:** 2/26/2022 10:29:32 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
122		ph.telegra.Telegraph		incoming message	2/26/2022 10:29:29 PM(UTC+0) **End time:** 2/26/2022 10:29:29 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
123		ph.telegra.Telegraph		outgoing message	2/26/2022 10:28:57 PM(UTC+0) **End time:** 2/26/2022 10:28:57 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
124		ph.telegra.Telegraph		outgoing message	2/26/2022 10:28:53 PM(UTC+0) **End time:** 2/26/2022 10:28:53 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
125		ph.telegra.Telegraph		outgoing message	2/26/2022 10:28:45 PM(UTC+0) **End time:** 2/26/2022 10:28:45 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
126		ph.telegra.Telegraph		outgoing message	2/26/2022 10:28:39 PM(UTC+0) **End time:** 2/26/2022 10:28:39 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
127		ph.telegra.Telegraph		incoming message	2/26/2022 10:28:27 PM(UTC+0) **End time:** 2/26/2022 10:28:27 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
128		ph.telegra.Telegraph		incoming message	2/26/2022 10:28:23 PM(UTC+0) **End time:** 2/26/2022 10:28:23 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
129		ph.telegra.Telegraph		incoming message	2/26/2022 10:28:20 PM(UTC+0) **End time:** 2/26/2022 10:28:20 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	
130		ph.telegra.Telegraph		outgoing message	2/26/2022 10:27:56 PM(UTC+0) **End time:** 2/26/2022 10:27:56 PM(UTC+0)	**PID:** 0 **TID:** 0 **Effective UID:** 0	**Source:** InteractionC	

Figure 12.9 – The Extraction Report Evidence list

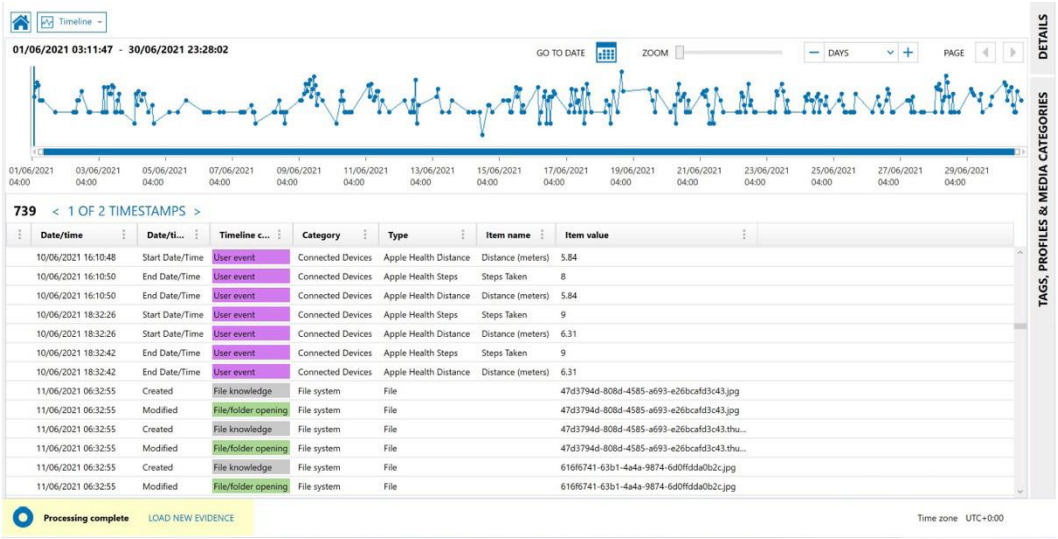


Figure 12.10 – A timeline displayed in Magnet AXIOM

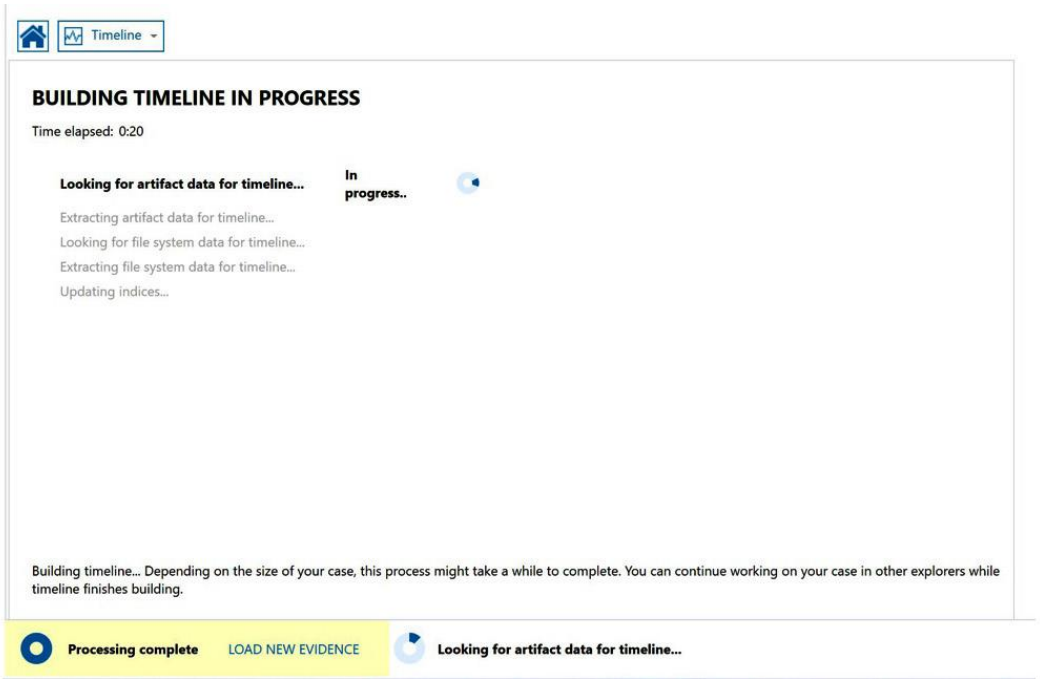


Figure 12.11 – Building a timeline in Magnet AXIOM

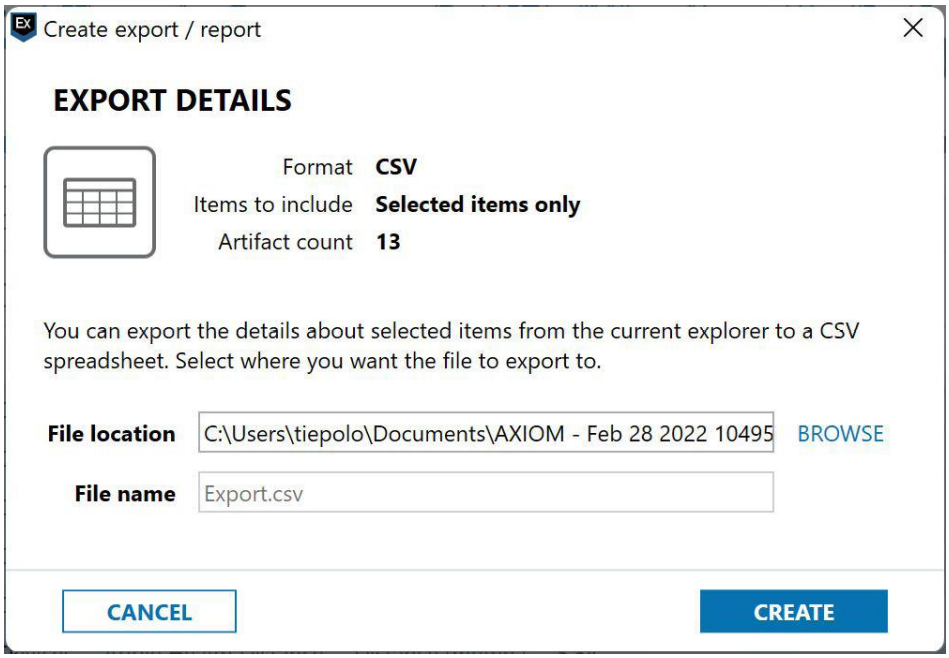


Figure 12.12 – Exporting the timeline to a CSV file

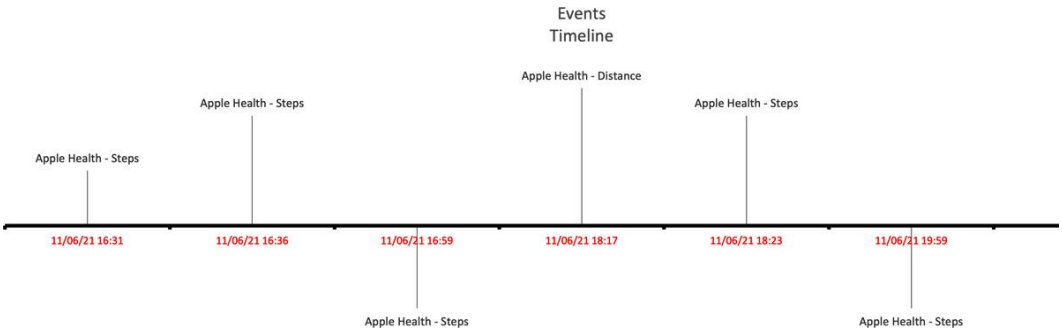


Figure 12.13 – A timeline created in Microsoft Excel