FOR585:

Smartphone Forensic Analysis In-Depth

MOST RELEVANT EVIDENCE PER GIGABYTE!

$25.00
DFPS_FOR585_v2.6_1-19

Poster Created by Heather Mahalik, Domenica "Lee" Crognale, and Cindy Murphy with support of the SANS DFIR Faculty

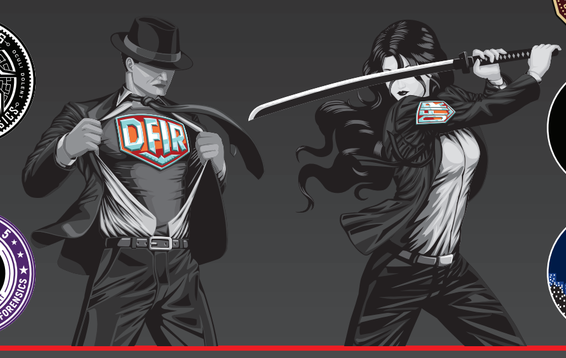
SANS DFIR
DIGITAL FORENSICS & INCIDENT RESPONSE

FOR500 Windows Forensics — GCFE
FOR518 Mac and iOS Forensic Analysis and Incident Response
FOR526 Advanced Memory Forensics & Threat Detection
FOR585 Smartphone Forensic Analysis In-Depth — GASF

OPERATING SYSTEM & DEVICE IN-DEPTH

INCIDENT RESPONSE & THREAT HUNTING

FOR508 Advanced Incident Response and Threat Hunting — GCFA
FOR572 Advanced Network Forensics: Threat Hunting, Analysis, and Incident Response — GNFA
FOR578 Cyber Threat Intelligence — GCTI
FOR610 REM: Malware Analysis — GREM
SEC504 Hacker Tools, Techniques, Exploits, and Incident Handling — GCIH



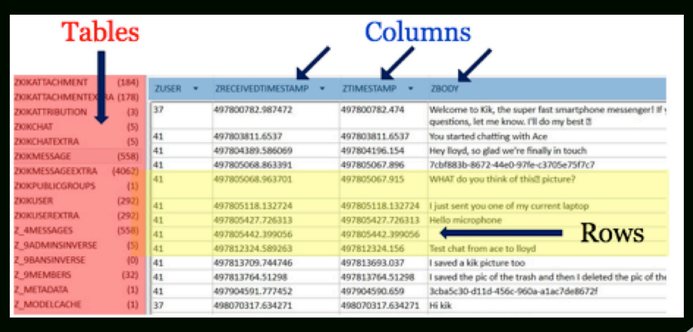
@sansforensics sansforensics dfir.to/DFIRCast dfir.to/MAIL-LIST

MOST RELEVANT EVIDENCE PER GIGABYTE!

SQLite

SQLite Database Basics

SQLite databases are a self-contained database stored as a file system file (but may have a few supporting files that will also be needed for analysis!) Files have the magic number "`SQLite format 3`." SQLite files correspond to a database that contains tables. `Tables` contain `rows` of data with corresponding `columns` that describe the data in the row.



Some temporary files may also be created, including `Journal files` and `Write Ahead Logs`. `Journal files` store original data before a transaction change so the database can be restored to a known state if an error occurs. They are created by default.

Write Ahead Logs (WAL) contain new data changes, leaving original database untouched. After a set number of page changes, the WAL is used to update the actual database. Write ahead logs are optional. Journal files – stores original data before a transaction change so the database can be restored to a known state if an error occurs (created by default).

Timestamp Conversion

Timestamps are stored in the databases as one of several numerical representations. *(Timestamps are assumed to be stored in UTC, you may need to verify this.)*

UNIX Epoch (10 digit number - number of seconds since 01/01/1970 00:00:00):

- `SELECT datetime(TS_COLUMN,'unixepoch')`
  Or in **local time** as suggested by the device settings (this can be done for all the following timestamps):
- `SELECT datetime(TS_COLUMN,'unixepoch', 'localtime')`

UNIX Epoch MILLISECONDS (13 digit number - number of milliseconds since 01/01/1970 00:00:00):

- `SELECT datetime(TS_COLUMN/1000,'unixepoch');`

Mac Absolute time, number of seconds since 01/01/2001 00:00:00. In order to correctly convert this timestamp, first, add the number of seconds since UNIXEPOCH time to Mac Absolute Time (978307200), then convert.

- `SELECT datetime(TS_COLUMN + 978307200, 'unixepoch');`

Chrome time accounts for time accurate to the MICROSECOND, which requires dividing the number by 1,000,000:

- `SELECT datetime(TS_COLUMN/1000000 + (strftime('%s','1601-01-01')),'UNIXEPOCH');`

Table Joins

Taking data from two (or more!) tables that have a column in common and joining them into one table. Identify tables of interest that contain unique values.

LEFT JOIN – Resulting rows are returned from the **LEFT** table even if there are no matches in the right. Using the **LEFT JOIN** produced all the text messages including those with and without attachments.

```
SELECT ZVIBERMESSAGE.ZTEXT AS "Message Text",
ZATTACHMENT.ZNAME AS "Attachment Filename",
datetime(ZVIBERMESSAGE.ZDATE+978307200,'unixepoch',
'localtime') AS "Message Date", ZVIBERMESSAGE.ZSTATE
AS "Message Direction/State" FROM ZVIBERMESSAGE LEFT
JOIN ZATTACHMENT on ZATTACHMENT.Z_PK=ZVIBERMESSAGE.
ZATTACHMENT
```

INNER JOIN - Resulting rows are returned when both items are a match. Using the **INNER JOIN** (also achieved by typing "**JOIN**" in the query) returned just the messages that included attachments.

Basic Analysis Query Structure

Get everything from a single table:
`SELECT * FROM A_TABLE;`

Get two columns from a single table:
`SELECT COLUMN_A, COLUMN_B FROM A_TABLE;`

Useful Stuff

Column Renaming:

A_TABLE.ZAWKWARDCOLUMNNAME AS "Chat Messages"

Counting:

```
SELECT COUNT(*) FROM A_TABLE;
```

Aggregating with GROUP BY and COUNT (Count chat messages per contact):

```
SELECT MESSAGES,COUNT(*) FROM CHAT GROUP BY CONTACT;
```

Sorting with ORDER BY:

```
SELECT * FROM CHAT ORDER BY A_TIMESTAMP ASC
```

ASC = Ascending DESC = Descending

Searching with WHERE and LIKE:

```
SELECT CONTACT, MESSAGE FROM CHAT WHERE CONTACT LIKE '%Hank%'
```

Mobile Malware and Spyware

Common Signs and Symptoms

- Android devices are most at risk for mobile malware infection
- Poor battery life
- Dropped calls and call disruptions
- Unusually large phone bills
- Data plan spikes
- Device performance problems
- Unexpected device behaviors
  - Unplanned reboots - Apps that close or open on their own - Unexplained settings changes
- Unexplained application errors
- High-risk user behavior
  - Risky downloads, browsing or link-clicking
- Spyware: Device was out of owner's control
  - Spyware installation requires possession of the device

Unpacking and Decompiling an Application File (.apk)

Prep:

- **INSTALL** most recent version of **Dex2Jar** on your desktop:
  http://code.google.com/p/dex2jar/downloads/list
- **INSTALL** most recent version of **JD-GUI** on your desktop:
  http://www.softpedia.com/get/Programming/Debuggers-Decompilers-Dissasemblers/JD-GUI.shtml
- **INSTALL** most recent **Java Development Kit**:
  http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Step 1:

- **RENAME** the application (.apk) file, appending a .zip extension to the end of the file name. EXAMPLE: **zombie_highway.apk** becomes **zombie_highway.apk.zip**

Step 2:

- **DOUBLE CLICk** on the newly named **.zip** file to open it and see the contents of the file.
- **LOCATE** the **classes.dex** file within the unzipped file.
- **COPY** the **classes.dex** file.

Step 3:

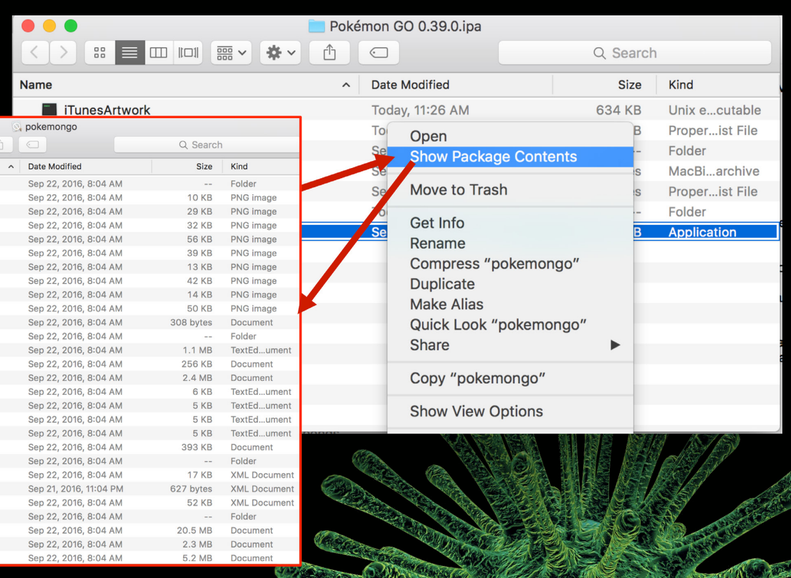
- **PASTE** the **classes.dex** file into the **dex2jar directory** created during prep stage.
- **OPEN** a command prompt and navigate to the **dex2jar directory** on the desktop.
- **EXECUTE** the batch file "`d2j-dex2jar.bat classes.dex`"
- This command will create a file named **classes_dex2jar.jar** in the **dex2jar directory**.

Step 4:

- **OPEN** the **jd-gui** Java Decompiler and navigate to the **classes_dex2jar.jar** created in the previous step.
- **OPEN** the **classes_dex2jar.jar** file to view and **NAVIGATE** the contents of the programming to reveal what the .apk file is doing.

Unzipped iOS – .ipa file

Unzipped iOS - .ipa file



Detection

- Finding malware
  ijcset.com/docs/IJCSET13-04-04-094.pdf

MOBILE MALWARE DETECTION

- SIGNATURE BASED
- SPECIFICATION BASED
- BEHAVIORAL BASED
- DATA MINING
- CLOUD BASED

Online Mobile Malware Sandboxes

- Upload suspicious .apk files to the Internet for online sandbox analysis:*
  - http://www.apk-analyzer.net
  - http://mobilesandbox.org
  - https://anubis.iseclab.org
  - https://code.google.com/p/droidbox

**Depending upon your location in the world, these sites may or may not be blocked.*

Local Static Malware Analysis

- Tools installed on local machine for mobile malware analysis:
  - Android SDK
  - Dex2Jar
  - Dexter
  - JD-GUI
- Virtual machine environments for mobile malware analysis:
  - Santoku

Mobile Malware Prevention

- Installation of mobile malware antivirus apps can assist users in preventing and detecting infection. Some mobile malware antivirus providers include:
  - Avast
  - AVG
  - BitDefender
  - Kaspersky
  - Lookout
  - Sophos
  - TrendMicro
  - Symantec (Norton)
  - TrustGo

ADB Commands

Requires USB Debugging be enabled

```
adb devices
adb shell pm list packages
adb shell service list
adb shell dumpsys <service of choice>
```

Example: wifi, usagestats, user, etc.

```
adb backup -all
```

Pay attention to the device – requires interaction

libimobiledevice

Should work on locked iOS devices, but may require a trust relationship

`ideviceinfo` provides device information including encrypted state, activation status, TimeZone, Phone Number, iOS version and more

`idevicepair pair` can be used to pair via CLI

`idevice_id.exe -l` provides the 40 digit GUID for the device

`idevicename` provides the name of the device

`idevicecrashreport -e <path for output>` contains traces of application usage

Smartphone Acquisition Tips

A Device On & Unlocked

- Logical/Backup Acquisition
- File System/Adv. Logical Acquisition
- Physical Acquisition, if supported
- Acquire SD and SIM card separately

B Device Locked (On or Off)

- **Only power on when directed by the tool**
- **Bypassing a lock does not bypass encryption**
- Physical Acquisition, if supported
- File system/Adv. Logical, Logical/Backup Acquisition
- Crack the lock, if possible
- Perform Advanced Acquisition
  - Chip-off • ISP/JTAG • Vendor Service (CAIS, GreyShift)
  - EDL/Bootloaders, root, jailbreaks

C Device Inaccessible

- Pull Cloud Data
  - Google, iCloud, Cloud Sync, etc.
- Search for smartphone backups
- Consider continuity and sync artifacts

DFIR Smartphone Forensics

MOST RELEVANT EVIDENCE PER GIGABYTE!

SANS FOR585:
Smartphone Forensic Analysis In-Depth

Course Authors

Heather Mahalik
hmahalik@gmail.com
@heathermahalik

Domenica Crognale
domenica.crognale@gmail.com
@domenicacrognal

Cindy Murphy
cindymurphy2412@gmail.com
@cindymurph

twitter.com/sansforensics

Common Smartphone Evidence Locations

Some of the artifacts listed for the iPhone and Android may be recoverable from all dumps or just physical access depending on the device."

Android

Partition	File	Description
Data	/com.android.providers.contacts/databases/contacts2.db /com.android.providers.contacts/databases/calllog.db /com.sec.android.provider.logsprovider/databases/Logs.db	Call logs Call logs (OS 7) Call logs and more!
Data	/system/accounts*.db	User account information
Data	/com.android.providers.contacts/ databases/contacts2.db /com.android.providers.contacts/ databases/contacts3.db	Contacts Contacts (OS 7)
Data	/com.android.providers.telephony/ databases/mmssms.db	SMS/MMS
Data	/com.google.android.apps.maps/*	Maps
Data	/com.sec.android.daemonapp/db/weatherClock	Location artifacts
Data	/com.google.android.gm/databases/<mail-name>.db	Gmail snippets
Data	/com.google.android.gms/databases/herrevad	Wireless and MAC addresses
Data	/system/locksettings.db and locksettings.db-WAL	Lock settings information
Data	/com.android.providers.settings/databases/settings.db and settings.db-WAL	Lock settings information
Data	/com.android.providers.media/external*.db and external*.db-WAL	Traces to SD card used in the device.
Data	/com.android.vending/databases/localappstate.db	Application traces
Data	/com.samsung.android.providers.context.databases. ContextLog_0.db (OS 7)	Application traces for Samsung devices
Data	/com.google.android.gms/databases/NetworkUsage.db /com.google.android.gms/databases/ns.db /com.google.android.gms/databases/reminders.db	Application, User and Location traces
Data	/com.android.providers.settings/*	Great place for username and passwords
Data	/system/*.key	Files needed for password cracking
Data	/system/device_policies.xml	Password requirements and policies.
Data	/system.SimCard.dat	Sim card and phone number information

Rooted Android Devices

Partition	File	Description
Data	/system/accounts*.db	User account information
Data	/com.google.android.gm/databases/<mail-name>.db	Gmail snippets
Data	/com.android.email/databases/EmailProvider.db	Email artifacts
Data	/com.google.android.gms/databases/herrevad	Wireless and MAC addresses
Data	/system/locksettings.db and locksettings.db-WAL	Lock settings information
Data	/com.android.providers.media/external*.db and external*.db-WAL	Traces to SD card
Data	/com.android.vending/databases/localappstate.db	Application traces
Data	/com.google.android.locations/files/cache.cell /com.google.android.locations/files/cache.wifi	Cellular and WiFi
Data	/com.samsung.android.providers.context.databases.ContextLog_0.db (OS 7)	Application traces for Samsung devices
Data	/com.google.android.gms/databases/NetworkUsage.db /com.google.android.gms/databases/ns.db /com.google.android.gms/databases/reminders.db	Application, User and Location traces
Data	/system/packages.xml /system/packages.list /system/netpolicy.xml	Application permissions
Data	/system/usagestats/0/<various directories>/*.xml	Application Usage
Data	/system/batterystats.bin /system/batterystats-daily.xml /system/batterystats-checkin.bin	Application Usage (may be difficult to parse)
Data	/com.sec.android.app.launcher/databases/launcher.db /com.android.providers.downloads/databases/downloads.db	Application artifacts (even after deleted)
Data	/system/dmappmgr.db	Application Usage
Data	/com.android.providers.settings/*	Great place for username and passwords
Data	/data/*	Application directories include more data
Data	/system/recent_images/*.png	Application snapshots may exist here

Jailbroken iOS Devices

Database	Description
/Library/CoreDuet/*	Device lock state (1=Locked, 0=Unlocked)
/Library/AggregateDictionary/ADDataStore.sqlitedb	Dictionary
/Library/BatteryLife/CurrentPowerLog.PLSQL	Battery life tracker, Application traces
/private/var/networkd/netusage.sqlite	Network artifacts
/Library/Health/healthdb.sqlite /Library/Health/healthdb_secure.sqlite	Activity, Personal information, more
/Library/Caches/com.apple.routined/cache_encrypted*.db /Library/Caches/com.apple.routined/StateModel*.archive	Frequent Locations (https://github.com/mac4n6/iOS-Frequent-Locations-Dumper)
/Library/Caches/cache_encrypted*.db /Library/Caches/lockCache_encrypted*.db	Cell and WiFi locations
/Applications/*	Examine relevant app directories to obtain additional data
/Library/BullitenBoard/ClearedSections.plist	Logs of cleared notifications
/Library/Keyboard/UserDictionary.sqlite	User created auto-correct
/Library/Accounts/Accounts3.sqlite	Accounts, user information, etc.
/Library/Databases/CellularUsage.db	SIMs used in device, including most recent
/Library/TCC/TCC.db	Applications permissions
/Library/Databases/Datausage.sqlite	Application traces
/Library/com.apple.itunesstored/itunesstored2.sqlitedb	Application traces

plist	Description
/Lockdown/device_values.plist	Activated state, BT address and more
/Preferences/com.apple.homesharing.plist	iCloud account information
/Preferences/com.apple.assistant.backedup.plist	Cloud sync settings
/Preferences/com.apple.coreduetd.plist	sync devices
com.apple.commcenter.plist	Device phone number, Network carrier, ICCIDs and IMSIs
com.apple.identityservices.idstatuscache.plist	iCloud sync, Email, FaceTime, Email, more
com.apple.accountsettings.plist	Email accounts pushed to device
com.apple.Maps.plist	Last latitude and longitude, map search history
/Library/Maps/Bookmarks.plist com.apple.Maps/Maps com.apple.Maps/Maps	Maps bookmarks History.mapsdata (iOS 7) GeoHistory.mapsdata (iOS 8 - iOS 11) *Pull cloud if possible
com.apple.MobileBluetooth.devices.plist	Synced devices
CloudConfigurationDetails.plist	Cloud configurations
/SystemConfiguration/com.apple.wifi.plist	WiFi
/SystemConfiguration/preferences.plist	WiFi and more
/Library/DataAccess/AccountInformation.plist	Email sync data
/Library/DataAccess/iCloud-[iCloud email account name]/*	iCloud Email account information and offline cache

Files of Interest	Description
/Library/Preferences/*	Examine plists for more information
/Library/DataAccess	Account information used to set up apps (Email, #, etc)
/var/mobile/Library/Keyboard	dynamic-text.dat

iOS Devices

Database	Description
Library/CallHistory/call_history.db Library/CallHistoryDB/CallHistory.storedata	Call logs Call record ((iOS 8 – iOS 10)
Library/AddressBook/AddressBook.sqlitedb	Contacts
Library/AddressBook/AddressBookImages.sqlitedb	Contact images
Library/SMS/sms.db	SMS messages
Library/SMS/Attachments/*	MMS file
Library/Calendar/Calendar.sqlitedb	Calendar
Library/Notes/notes.sqlite	Notes
Library/Safari/*	Safari activity
Library/Accounts/Accounts3.sqlite	Account information
Library/BulletinBoard/ClearedSections.plist	Logs of cleared notifications
Media/PhotoData/Photos.sqlite	Metadata about multimedia files
Library/TCC/TCC.db	Application permissions
Library/Databases/DataUsage.sqlite	Application information and usage details
Library/ADDataStore.sqlite	iOS unlock data repository (Refer to mac4n6.com)
Library/CoreDuet/coreduetd.db	unlock data repository (Refer to mac4n6.com)

plist	Description
com.apple.commcenter.plist	Device phone number, network carrier, ICCIDs, and IMSIs
com.apple.accountsettings.plist	Email accounts pushed to device
com.apple.Maps.plist	Last latitude and longitude, map search history
Library/Maps/Bookmarks.plist com.apple.Maps/Maps com.apple.Maps/Maps	Maps bookmarks History.mapsdata (iOS 7) GeoHistory.mapsdata (OS 8 – iOS 10)
SystemConfiguration/com.apple.wifi.plist	WiFi
SystemConfiguration/preferences.plist	WiFi and more
Library/Preferences/com.apple.mobilenotes.plist	Notes
Library/SpringBoard/IconState.plist	Homescreen icon layout
Library/ConfigurationProfiles/UserSettings.plist	User-created restrictions
Library/Preferences/com.apple.springboard.plist	User-created restrictions
Library/Preferences/com.apple.WebFoundation.plist	Safari activity
Library/Preferences/com.apple.MobileSMS.plist	SMS, iMessage and FaceTime
Library/Preferences/com.apple.madrid.plist	SMS, iMessage and FaceTime
Library/DataAccess/AccountInformation.plist	Email sync data
Library/DataAccess/iCloud-'iCloud email account name'/.mboxCache.plist	iCloud email account information
Library/DataAccess/iCloud-'iCloud email account name'/.OfflineCache/'number'	iCloud offline cache

FOR585: Smartphone Forensic Analysis In-Depth

A smartphone lands on your desk and you are tasked with determining if the user was at a specific location at a specific date and time. You rely on your forensic tools to dump and parse the data. The tools show location information tying the device to the place of interest. Are you ready to prove the user was at that location? Do you know how to take this further to place the subject at the location of interest at that specific date and time? Tread carefully, because the user may not have done what the tools are showing!"

SMARTPHONE DATA CAN'T HIDE FOREVER –
IT'S TIME TO OUTSMART THE MOBILE DEVICE!

Poster Created by Heather Mahalik, Domenica "Lee" Crognale, and Cindy Murphy