The Unified Logging Confession
Agenda

Case Issue -> Acquisition -> Analysis
Speakers

Johann POLEWCZYK
Mac Forensic expert of the French Gendarmerie National Lab since 2007
Digital Forensic expert at School of Criminal Sciences (University of Lausanne) since 15 days

Matthieu REGNERY
Was Head of the Data Extraction Unit of the French Gendarmerie National Lab
New job since August 2019
Present in the conference room to answer your questions
CASE ISSUE
• Context

▶ Criminal case

- Time frame of the facts quickly narrowed down by the investigators
- Suspect
  - Owner of an iPhone 6s (iOS 10.3.3)
  - Movements proven during the factual period
  - Phone went off the mobile carrier several times

- Suspect explains that his cellphone:
  - Does not receive the network very well
  - Has a battery problem and regularly turns off
• Context

▷ Mission
  ◦ Analyze the phone to confirm the suspect’s explanations or not.

▷ Analysis
  ◦ Different forensics tools available on the market were used:
    - Unable to provide information to answer the questions asked

▷ Question
  ◦ Is it really possible to find this kind of information within the phone memory?
• Parts of the answer

  ▶ Apple Event in Septembre 2017
    ○ Introduction of iPhone X
      - Phone unlocking problems with FaceID
Eléments de réponses

Apple Event de Septembre 2017

Présentation de l'iPhone X et FaceID
• Parts of the answer

▶ Apple Event in Septembre 2017
  ○ Introduction of iPhone X
    - Phone unlocking problems with FaceID

▶ Explanations provided by Apple
  ○ After analysis of the mobile phone logs, we were able to determine that just before the presentation on stage, people took in hand the phone. Their face was not enrolled with Face ID, the phone counted unsuccessful unlock attempts, prompting the user to enter the passcode

▶ New data extraction methods
  ○ FileSystem extraction (all visible files of data partition)
• Where to find this information?

  ▶ Search by keywords
    ◦ Airplane
    ◦ Battery
    ◦ Boot
    ◦ Shutdown

  ▶ Identification of searched information:
    ◦ SQLite database (knowledgeC.db)
    ◦ Logs files
    ◦ Binary files (Unified logs)
• Unified Logs

  ▶ WWDC 2016
    ☐ Introduction of a new logging system (Unified Logs, Tracing Activity)
    ☐ Introduced since macOS 10.12, iOS 10.0, tvOS 10.0 et watchOS 3.0

  ▶ Principle
    ☐ Data saved in compressed binary files (tracev3 format)
    ☐ Location:
      - /private/var/db/diagnostics
      - /private/var/db/uuidtext
    ☐ Logs portability:
      - Bundle in logarchive format
DATA ACQUISITION
• Different extraction methods available

▶ Need an unlocked device

▶ File System extraction (All visible files of data partition)
  ◦ Cellebrite: CAS (Cellebrite Advanced Services), UFED Premium
  ◦ Grayshift: GrayKey
  ◦ Jailbroken device
• Extraction methods available

▪ Keys combination on the mobile phone
  ◦ Volume down + Volume up + Side button
    - Slight vibration of the phone
    - Wait for the generation of the log files
      ✦ may take several minutes
      ✦ no progress indicator

  ◦ Accessibility: Assistive Touch (Analysis option)

▪ Then sync with iTunes
Accessibility: AssistiveTouch

AssistiveTouch allows you to use your iPhone if you have difficulty touching the screen or if you require an adaptive accessory.

Customise Top Level Menu...

CUSTOM ACTIONS

Single-Tap Open Menu

Double-Tap None

Long Press None

3D Touch Home

Custom actions allow you to interact directly with the AssistiveTouch icon without opening the menu.

Idle Opacity 40%

CUSTOM GESTURES

Double-Tap

Shake

App Switcher

Screenshot

Lock Rotation

Pinch

3D Touch

Double-Tap

SOS

Analytics

Reachability

Restart

Apple Pay

Speak Screen
AssistiveTouch

AssistiveTouch allows you to use your iPhone if you have difficulty touching the screen or if you require an adaptive accessory.

Customise Top Level Menu...

CUSTOM ACTIONS

Single-Tap  Open Menu
Double-Tap  Analytics
Long Press  None
3D Touch  Home

Custom actions allow you to interact directly with the AssistiveTouch icon without opening the menu.

Idle Opacity  40 %

CUSTOM GESTURES
Musique

Les morceaux et les vidéos que vous ajoutez à iTunes apparaissent dans votre bibliothèque musicale. La musique achetée sur iCloud apparaît également lorsque vous vous connectez à l’iTunes Store.

Aller dans l’iTunes Store  Se connecter à l’iTunes Store
DATA ANALYSIS
• Unified logs
  ▶ About 3 weeks of activity
  ▶ 600~700 Mb approx.
  ▶ 13 Millions of events
• Prerequisite

▷ Apple Computer
  • Operating system
    - iOS 10.x —> macOS 10.12 (Sierra) and above
    - iOS 11.x —> macOS 10.13 (High Sierra) and above
    - iOS 12.x —> macOS 10.14 (Mojave) and above
    - iOS 13.x —> macOS 10.15 (Catalina)

  • If FileSystem extraction
    - logarchive Bundle creation
      ✦ Copy content of /private/var/db/diagnostics and /private/var/db/uuid in a new folder
      ✦ Rename it with logarchive extension

• Console App

• Terminal
  - log command
• Console App
  ▸ Advantages
    ◦ Graphical User Interface
    ◦ Usability
  ▸ Limits
    ◦ Missing events
    ◦ Opening logarchive bundles may take several minutes
<table>
<thead>
<tr>
<th>Type</th>
<th>Date et heure</th>
<th>Processus</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1970-01-01 01:15:39.336892</td>
<td>kernel</td>
<td>iBoot version: iBoot-4076.50.114</td>
<td></td>
</tr>
<tr>
<td>1970-01-01 01:15:39.797964</td>
<td>kernel</td>
<td>IOTimeSyncClockManager::init created and initing</td>
<td></td>
</tr>
<tr>
<td>1970-01-01 01:15:39.788876</td>
<td>kernel</td>
<td>AUC&lt;private&gt;::init&lt;private&gt;</td>
<td></td>
</tr>
<tr>
<td>1970-01-01 01:15:39.798116</td>
<td>kernel</td>
<td>AUC&lt;private&gt;::probe&lt;private&gt;,&lt;private&gt;</td>
<td></td>
</tr>
<tr>
<td>1970-01-01 01:15:39.788677</td>
<td>kernel</td>
<td>ACMDRMAalytics: init: called,</td>
<td></td>
</tr>
<tr>
<td>1970-01-01 01:15:39.799965</td>
<td>kernel</td>
<td>IOTimeSyncClockManager::start starting</td>
<td></td>
</tr>
<tr>
<td>1970-01-01 01:15:39.799238</td>
<td>kernel</td>
<td>AUC&lt;private&gt;::start&lt;private&gt;</td>
<td></td>
</tr>
<tr>
<td>1970-01-01 01:15:39.713189</td>
<td>kernel</td>
<td>AppleS8000ID::start: chip-revision: 08</td>
<td></td>
</tr>
<tr>
<td>1970-01-01 01:15:39.713517</td>
<td>kernel</td>
<td>AppleS8000ID::start: this: &lt;private&gt;, TCC virt addr: &lt;private&gt;, TCC phys addr: 0x2022400000</td>
<td></td>
</tr>
<tr>
<td>1970-01-01 01:15:39.717377</td>
<td>kernel</td>
<td>AppleARMBacklight::start: Using new Backlight Architecture 1</td>
<td></td>
</tr>
<tr>
<td>1970-01-01 01:15:39.717418</td>
<td>kernel</td>
<td>AppleARMBacklight::start: no DBV offset data</td>
<td></td>
</tr>
<tr>
<td>1970-01-01 01:15:39.717436</td>
<td>kernel</td>
<td>AppleARMBacklight::start: no DBV max data</td>
<td></td>
</tr>
</tbody>
</table>
• Terminal: log command

  ▶ Advantages
  ◦ Speed
  ◦ Multiple filters
  ◦ Export in csv, json, syslog

  ▶ Limits
  ◦ Command Line (many people are afraid by the Terminal)
• Terminal: log command

usage:
    log <command>

global options:
    -, --help
    -q, --quiet
    -v, --verbose

commands:
    collect  gather system logs into a log archive
    config   view/change logging system settings
    erase    delete system logging data
    show     view/search system logs
    stream   watch live system logs
    stats    show system logging statistics

further help:
    log help <command>
    log help predicates _
2018-02-14 01:25:22.012353+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] Stats:

2018-02-14 01:25:22.012390+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 0: 0
2018-02-14 01:25:22.012427+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 1: 0
2018-02-14 01:25:22.012463+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 2: 0
2018-02-14 01:25:22.012500+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 3: 0
2018-02-14 01:25:22.012538+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 4: 0
2018-02-14 01:25:22.012574+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 5: 10388
2018-02-14 01:25:22.012610+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 6: 0
2018-02-14 01:25:22.012646+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 7: 0
2018-02-14 01:25:22.012683+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 8: 0
2018-02-14 01:25:22.012720+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 9: 0
2018-02-14 01:25:22.012756+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 10: 0
2018-02-14 01:25:22.012793+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 11: 0
2018-02-14 01:25:22.012823+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 12: 0
2018-02-14 01:25:22.012864+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 13: 0
2018-02-14 01:25:22.012905+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 14: 0
2018-02-14 01:25:22.012942+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 15: 0
2018-02-14 01:25:22.012978+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 16: 0
2018-02-14 01:25:22.012981+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 17: 0
2018-02-14 01:25:22.012985+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] 18: 0

kernel: (AppleBiometricSensor) handleCMDSync() entry: cmd=0x8d, param=0x0, param1=0x0, param2=0x0

2018-02-14 01:25:22.012944+0100 0x1bd  Default 0x0  0  0 kernel: [AppleBiometricSensor] handleInterrupt() entry: state=1

2018-02-14 01:25:22.012338+0100 0x1bd  Default 0x0  0  0 kernel: [AppleBiometricSensor] Assert Macros: result == 0 (value = 0xffffffff ffe00002c7), unsupported function file: /BuildRoot/Library/Caches/com.apple.xbs/Sources/AppleBiometricSensor/AppleBiometricSensor-192.50.3/AppleMesa.cpp, line: 298

error from handleInterrupt (unsupported function)

param1=0x0, param2=0x0

2018-02-14 01:25:22.023498+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] handleCMDSync() entry: cmd=0x15, param=0x0, param1=0x0, param2=0x0

2018-02-14 01:25:22.023573+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] powerOnReset() entry

2018-02-14 01:25:22.023689+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] handleInterrupt() entry: state=1

2018-02-14 01:25:22.023859+0100 0x6a73  Default 0x0  0  0 kernel: [AppleBiometricSensor] ResetAction() entry: _client=<private>, isOn()=1

2018-02-14 01:25:22.023946+0100 0x6a852  Default 0x0  0  0 kernel: [AppleMesaSEPDiver] HandleReset <- status:<private>, statusSize:16

, resetReason:2 (_sensorOpen:1, _sensorState:0)

2018-02-14 01:25:22.023989+0100 0x6a852  Default 0x0  0  0 kernel: [AppleMesaSEPDiver] HandleReset: kResetReasonESD

2018-02-14 01:25:22.023237+0100 0x6a852  Default 0x0  0  0 kernel: [AppleMesaSEPDiver] LoadPatch <- patch:0x0

2018-02-14 01:25:22.023253+0100 0x6a852  Default 0x0  0  0 kernel: [AppleMesaSEPDiver] handleCMDSync() entry: cmd=0x0, param=0x0, param1=0x0, param2=0x0

2018-02-14 01:25:22.023281+0100 0x1bd  Default 0x0  0  0 kernel: [AppleBiometricSensor] handleInterrupt() entry: state=1

2018-02-14 01:25:22.023438+0100 0x6dc  Default 0x0  0  0 kernel: [AppleBiometricSensor] biometricKeyId: (lib8KMDMdylib) [com.apple.biometrickit:Daemon-Mesa] statusMessage: withData:timestamp: 84 @ 71258829896

2018-02-14 01:25:22.023824+0100 0x6a852  Default 0x0  0  0 kernel: [AppleMesaSEPDiver] prepareSession: Establishing UID session...

2018-02-14 01:25:22.023829+0100 0x6a852  Default 0x0  0  0 kernel: [AppleMesaSEPDiver] performKeyExchange <- sessionType:0

2018-02-14 01:25:22.023940+0100 0x6a852  Default 0x0  0  0 kernel: [AppleBiometricSensor] handleCMDSync() entry: cmd=0x0e, param=0x028, param1=<private>, param2=0x0
usage: log show [options] <archive>
or: log show [options]

description:
Show the contents of the system log datastore or a log archive.
Output contains only default level messages unless --info and/or
--debug are specified.

options:
--[no-]backtrace Control whether backtraces are shown
--[no-]debug Control whether "Debug" events are shown
--[no-]info Control whether "Info" events are shown
--[no-]loss Control whether message loss events are shown
--[no-]signpost Control whether signposts are shown
--color <mode> Control color output (valid: auto, always, none)
--end <date> Display events up to the given end date
--last <num>[m|h|d] Display recent events up to the given limit
--predicate <predicate> Filter events using the given predicate
--source Annotate output with source file and line-number
--start <date> Display events from the given start date
--style <style> Output format (valid: syslog, json, compact)
--timezone local | <tz> Use the given timezone when displaying event timestamps
--mach-continuous-time Print mach continuous time timestamps rather than walltime

valid time formats:
<table>
<thead>
<tr>
<th>Predicate</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>activityIdentifier</td>
<td>(integer)</td>
</tr>
<tr>
<td>bootUUID</td>
<td>(uuid)</td>
</tr>
<tr>
<td>category</td>
<td>(string)</td>
</tr>
<tr>
<td>composedMessage</td>
<td>(string)</td>
</tr>
<tr>
<td>continuousNanosecondsSinceBoot</td>
<td>(integer)</td>
</tr>
<tr>
<td>creatorActivityIdentifier</td>
<td>(integer)</td>
</tr>
<tr>
<td>creatorProcessUniqueIdentifier</td>
<td>(integer)</td>
</tr>
<tr>
<td>date</td>
<td>(date)</td>
</tr>
<tr>
<td>formatString</td>
<td>(string)</td>
</tr>
<tr>
<td>logType</td>
<td>(log type)</td>
</tr>
<tr>
<td>machContinuousTimestamp</td>
<td>(integer)</td>
</tr>
<tr>
<td>parentActivityIdentifier</td>
<td>(integer)</td>
</tr>
<tr>
<td>process</td>
<td>(string)</td>
</tr>
<tr>
<td>processIdentifier</td>
<td>(integer)</td>
</tr>
<tr>
<td>processImagePath</td>
<td>(string)</td>
</tr>
<tr>
<td>processImageUUID</td>
<td>(uuid)</td>
</tr>
<tr>
<td>sender</td>
<td>(string)</td>
</tr>
<tr>
<td>senderImageOffset</td>
<td>(integer)</td>
</tr>
<tr>
<td>senderImagePath</td>
<td>(string)</td>
</tr>
<tr>
<td>senderImageUUID</td>
<td>(uuid)</td>
</tr>
<tr>
<td>signpostIdentifier</td>
<td>(integer)</td>
</tr>
<tr>
<td>signpostScope</td>
<td>(signpost scope)</td>
</tr>
<tr>
<td>signpostType</td>
<td>(signpost type)</td>
</tr>
<tr>
<td>size</td>
<td>(integer)</td>
</tr>
<tr>
<td>subsystem</td>
<td>(string)</td>
</tr>
<tr>
<td>threadIdentifier</td>
<td>(integer)</td>
</tr>
<tr>
<td>timeToLive</td>
<td>(integer)</td>
</tr>
<tr>
<td>transitionIdentifier</td>
<td>(integer)</td>
</tr>
<tr>
<td>transitionActivityIdentifier</td>
<td>(integer)</td>
</tr>
<tr>
<td>type</td>
<td>(event type)</td>
</tr>
</tbody>
</table>
• Terminal: log command

  ▶ Usage

  log show log.logarchive --predicate 'composedMessage contains "[SearchedTerms]"' --start 'start_date' --end 'end_date'
• Terminal: log command

  ▪ Example

    log show log.logarchive --predicate 'composedMessage contains "Animating backlight to factor "' --start '2017-08-09' --end '2017-08-10'
<table>
<thead>
<tr>
<th>Time</th>
<th>Event Description</th>
<th>Process Name</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-02-07 18:46:01.776456</td>
<td>Terminal: log command</td>
<td>Data analysis</td>
<td></td>
</tr>
</tbody>
</table>
• Terminal: log command

SpringBoard: [com.apple.SpringBoard:Backlight] Animating backlight to factor 1.00 with duration 0.00 source:13 (prox)
SpringBoard: [com.apple.SpringBoard:Backlight] Animating backlight to factor 0.05 with duration 0.50 source:8 (idle timer)

SpringBoard: [com.apple.SpringBoard:Backlight] Animating backlight to factor 1.00 with duration 0.00 source:13 (prox)

SpringBoard: [com.apple.SpringBoard:Backlight] Animating backlight to factor 0.05 with duration 0.50 source:8 (idle timer)
• Using a DB

  • logdump can export all the log in a more scriptable format (JSON)
    ✴ Library/Developer/CommandLineTools/usr/bin/logdump --archive system_logs.logarchive --style json --source > myphonelogs.json

  • Insert in a mongodb
    ✴ mongodb/bin/mongoimport --db ioslogs --collection logs --file myphonelogs.json --jsonArray

  • query the db, script, plot
• Kind of data that can be extracted

▶ Supply
  ◦ Power on
    - kernel: iBoot version: iBoot-version
  ◦ Wake
    - SpringBoard: [com.apple.SpringBoard.Backlight] Animating backlight to factor 1.00 with duration 0.09 source:2 (home button)
  ◦ Battery percentage
    - coreduetd: [com.apple.coreduet.admissionCheck.] BatteryConditionChange: device:0, batteryLevel:92, isConnectedToCharger:0
  ◦ Power supply connexion
    - powerd: Power Source change. External connected:1
  ◦ Sleep mode
    - SpringBoard: [com.apple.SpringBoard.Backlight] Animating backlight to factor 0.00 with duration 0.18 source:8 (idle timer)
• Kind of data that can be extracted

> Unlocking

- **Passcode**
  - SpringBoard: (SpringBoardFoundation) [com.apple.SpringBoard.auth.Keybag] Keybag state changed: <SBFMobileKeyBagState: 0x1c020ee90; lockState: Unlocked; isEffectivelyLocked: NO; permanentlyBlocked: NO; recoveryRequired: NO; recoveryPossible: YES; shouldWipe: NO>

- **Touch ID / Face ID**
  - SpringBoard: (SpringBoardFoundation) [com.apple.SpringBoard.auth.Keybag] Keybag state changed: <SBFMobileKeyBagState: 0x1c020ee90; lockState: InBioUnlock; isEffectivelyLocked: NO; permanentlyBlocked: NO; recoveryRequired: NO; recoveryPossible: YES; shouldWipe: NO>
• Kind of data that can be extracted

  ▶ Screen state
  • Off
    - SpringBoard: [com.apple.SpringBoard.Backlight] Animating backlight to factor 0.00 with duration 0.18 source:8 (idle timer)
    - symptomsd: (SymptomEvaluator) [com.apple.symptomsd.basic] Power: screen is dark
    - powerd: Display state: Off
  • On (source event: notification, buttons...)
    - SpringBoard: [com.apple.SpringBoard.Backlight] Animating backlight to factor 1.00 with duration 0.09 source:2 (home button)
    - symptomsd: (SymptomEvaluator) [com.apple.symptomsd.basic] Power: screen is not dark
    - powerd: Display state: On
  • Touched
    - SpringBoard: (FrontBoard) [com.apple.FrontBoard.SystemGesture] immediate edge swipe: failed
• Kind of data that can be extracted
  
  Orientation
    🌳 SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from,
    Portrait -> LandscapeRight
    - Portrait
    - PortraitUpsideDown
    - LandscapeLeft
    - LandscapeRight
    - FaceUp
    - FaceDown
• Kind of data that can be extracted

▷ Airplane Mode
  ◦ Activation (Control Center, Settings, Siri)
    - SpringBoard: (ControlCenterUI) -[CCUIAirplaneModeSetting _toggleState] ENABLED
    - timed: [com.apple.timed.text] Airplane mode changed 0->1
  ◦ Deactivation
    - SpringBoard: (ControlCenterUI) -[CCUIAirplaneModeSetting _toggleState] DISABLED
    - timed: [com.apple.timed.text] Airplane mode changed 1->0
• Kind of data that can be extracted
  ▶ Network Events
    ◦ Signal force
    ◦ SIM card (Insertion, ejection)
    ◦ Wi-Fi
  ▶ Applications
    ◦ Usage
    ◦ Background
    ◦ Specific events
  ▶ Photos App
    ◦ Displaying
    ◦ Deletion
    ◦ Buttons
<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:45:57.161</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Unexpected -&gt; Portrait</td>
<td>Orientation: Portrait</td>
</tr>
<tr>
<td>06:47:29.771</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Ambiguous -&gt; Portrait</td>
<td>Orientation: Portrait</td>
</tr>
<tr>
<td>06:48:49.343</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Unexpected -&gt; FaceUp</td>
<td>Orientation: Face Up</td>
</tr>
<tr>
<td>06:49:37.339</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Ambiguous -&gt; Portrait</td>
<td>Orientation: Portrait</td>
</tr>
<tr>
<td>06:49:41.462</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Portrait -&gt; LandscapeLeft</td>
<td>Orientation: From Portrait to Landscape</td>
</tr>
<tr>
<td>06:49:42.154</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, LandscapeLeft -&gt; Portrait</td>
<td>Orientation: From Landscape to Portrait</td>
</tr>
<tr>
<td>06:49:54.755</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Unexpected -&gt; Portrait</td>
<td>Orientation: Face Up</td>
</tr>
<tr>
<td>06:49:55.962</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, FaceUp -&gt; LandscapeLeft</td>
<td>Orientation: From FaceUp to Landscape</td>
</tr>
<tr>
<td>06:49:59.597</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, LandscapeLeft -&gt; Portrait</td>
<td>Orientation: From Landscape to Portrait</td>
</tr>
<tr>
<td>06:51:00.726</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Unexpected -&gt; Portrait</td>
<td>Orientation: Face Up</td>
</tr>
<tr>
<td>06:51:01.834</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, FaceUp -&gt; Portrait</td>
<td>Orientation: From FaceUp to Portrait</td>
</tr>
<tr>
<td>06:51:32.324</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Unexpected -&gt; Portrait</td>
<td>Orientation: Portrait</td>
</tr>
<tr>
<td>06:51:32.750</td>
<td>coreeduSet: [com.apple.coreeduSet.admissionCheck.] BatteryConditionChange: device 0, batteryLevel: 92, isConnectedToCharger:0</td>
<td>Battery Level: 92%</td>
</tr>
<tr>
<td>06:54:33.739</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Unexpected -&gt; Portrait</td>
<td>Orientation: Portrait</td>
</tr>
<tr>
<td>06:55:30.367</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Unexpected -&gt; Portrait</td>
<td>Orientation: Portrait</td>
</tr>
<tr>
<td>07:00:01.191</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Unexpected -&gt; Portrait</td>
<td>Orientation: Portrait</td>
</tr>
<tr>
<td>07:06:10.070</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Unexpected -&gt; Portrait</td>
<td>Orientation: Portrait</td>
</tr>
<tr>
<td>07:10:05.179</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Unexpected -&gt; FaceUp</td>
<td>Orientation: Face Up</td>
</tr>
<tr>
<td>07:10:05.582</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, FaceUp -&gt; Portrait</td>
<td>Orientation: From FaceUp to Portrait</td>
</tr>
<tr>
<td>07:10:08.801</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, Portrait -&gt; FaceUp</td>
<td>Orientation: From Portrait to FaceUp</td>
</tr>
<tr>
<td>07:10:09.322</td>
<td>SpringBoard: (com.apple.SpringBoard.Backlight) Animating backlight to factor 1.00 with duration 0.09 source:2 (home button)</td>
<td>Wake up (Touch ID sensor)</td>
</tr>
<tr>
<td>07:10:09.384</td>
<td>symptomsd: [SymptomEvaluator] [com.apple.symptomsd.analytics] Power: screen is not dark</td>
<td>Screen is on</td>
</tr>
<tr>
<td>07:10:09.552</td>
<td>SpringBoard: (SpringBoardFoundation) [com.apple.SpringBoard.auth.Keybag] Keybag state changed:</td>
<td>Device unlocked with TouchID</td>
</tr>
<tr>
<td></td>
<td>&lt;SUFMobileKeyBagState: 0x1c020ee90; lockState: InBIOUnlock; isEffectivelyLocked: NO; permanentlyBlocked: NO; recoveryRequired: NO; recoveryPossible: YES, shouldWipe: NO&gt;</td>
<td></td>
</tr>
<tr>
<td>07:10:09.663</td>
<td>SpringBoard: (CoreMotion) [com.apple.locationd.Motion.Orientation] Notify from, FaceUp -&gt; Portrait</td>
<td>Orientation: From Face Up to Portrait</td>
</tr>
<tr>
<td>07:10:11.395</td>
<td>SpringBoard: (FrontBoard) [com.apple.FrontBoard.SystemGesture] Immediate edge swipe: failed</td>
<td>Screen touched</td>
</tr>
<tr>
<td>07:10:12.349</td>
<td>SpringBoard: (FrontBoard) [com.apple.FrontBoard.SystemGesture] Immediate edge swipe: failed</td>
<td>Screen touched</td>
</tr>
<tr>
<td>Time</td>
<td>Event interpretation</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>03:21:31.546</td>
<td>Orientation: Face Down</td>
<td></td>
</tr>
<tr>
<td>03:25:51.476</td>
<td>Wake Up (TouchID sensor)</td>
<td></td>
</tr>
<tr>
<td>03:25:51.498</td>
<td>Screen On</td>
<td></td>
</tr>
<tr>
<td>03:25:51.659</td>
<td>Orientation: Face UP</td>
<td></td>
</tr>
<tr>
<td>03:25:52.560</td>
<td>Orientation: Portrait</td>
<td></td>
</tr>
<tr>
<td>03:25:52.762</td>
<td>Orientation: Face UP</td>
<td></td>
</tr>
<tr>
<td>03:25:53.760</td>
<td>Orientation: Portrait</td>
<td></td>
</tr>
<tr>
<td>03:25:54.260</td>
<td>Orientation: Face UP</td>
<td></td>
</tr>
<tr>
<td>03:25:57.309</td>
<td>Displaying Control Center</td>
<td></td>
</tr>
<tr>
<td>03:25:58.132</td>
<td>Disabling Airplane Mode via Control Center button</td>
<td></td>
</tr>
<tr>
<td>03:25:58.167</td>
<td>Airplane Mode disabled</td>
<td></td>
</tr>
<tr>
<td>03:25:58.360</td>
<td>Orientation: Portrait</td>
<td></td>
</tr>
<tr>
<td>03:25:58.636</td>
<td>&quot;happn&quot; App: loading data in background</td>
<td></td>
</tr>
<tr>
<td>03:25:59.010</td>
<td>Closing Control Center</td>
<td></td>
</tr>
<tr>
<td>03:25:59.160</td>
<td>Orientation: Face UP</td>
<td></td>
</tr>
<tr>
<td>03:25:59.760</td>
<td>Orientation: Portrait</td>
<td></td>
</tr>
<tr>
<td>03:26:00.502</td>
<td>&quot;Mail&quot; App: loading data in background</td>
<td></td>
</tr>
<tr>
<td>03:26:00.660</td>
<td>Orientation: Face UP</td>
<td></td>
</tr>
<tr>
<td>03:26:01.060</td>
<td>Orientation: Portrait</td>
<td></td>
</tr>
<tr>
<td>03:26:01.110</td>
<td>Sleep mode by pressing side button</td>
<td></td>
</tr>
<tr>
<td>03:26:01.229</td>
<td>&quot;happn&quot; App: Loading geolocation data</td>
<td></td>
</tr>
<tr>
<td>03:26:01.256</td>
<td>&quot;happn&quot; App: Loading geolocation data</td>
<td></td>
</tr>
<tr>
<td>03:26:01.625</td>
<td>Screen Is Off</td>
<td></td>
</tr>
</tbody>
</table>
• Example: battery usage and charger plugged in
TAKEAWAYS

• Lots of data available

• Several possibilities to collect (need unlocked device)

• Information to answer key questions
  ▶ Forensics
    ◦ Correlating events/activity with owner hearing
    ◦ Verifying phone interactions (wifi, carrier, battery)
    ◦ Examining last actions (apps, orientation)

  ▶ Incident response
    ◦ Malware activity
    ◦ Suspicious app
    ◦ Connected devices
WHAT’S NEXT

‣ Growing demand especially in the context of fatal traffic accidents

‣ Implementing in forensic software solutions

‣ Improving knowledge on
  ◦ Event missing with with iTunes sync
  ◦ Differences in EventMessages contents based on iOS version
  ◦ Undiscovered events
Contacts

Johann POLEWCZYK
johann.polewczyk@unil.ch

Digital Forensic Department
inl.ircgn@gendarmerie.interieur.gouv.fr

Ecole des sciences criminelles

INSTITUT DE RECHERCHE CRIMINELLE
DE LA GENDARMERIE NATIONALE