REALITYNET
System Solutions

iOS FORENSICS: WHERE ARE WE NOW AND WHAT ARE WE MISSING?

MATTIA EPIFANI – PASQUALE STIRPARO

SANS EU DIGITAL FORENSICS SUMMIT 2016

PRAGUE, 9 OCTOBER 2016

AGENDA

- iOS acquisition challenges
- Search and seizure of iOS Devices
- Acquisition techniques
- Alternative options
- Application analysis

iOS ACQUISITION CHALLENGES

- iOS devices use **full disk encryption**
- Other protection layers (i.e. per-file key, backup password)
- **JTAG** ports are **not available**
- Chip-off techniques are not useful because of full disk encryption
  - But some experimental techniques are just out!

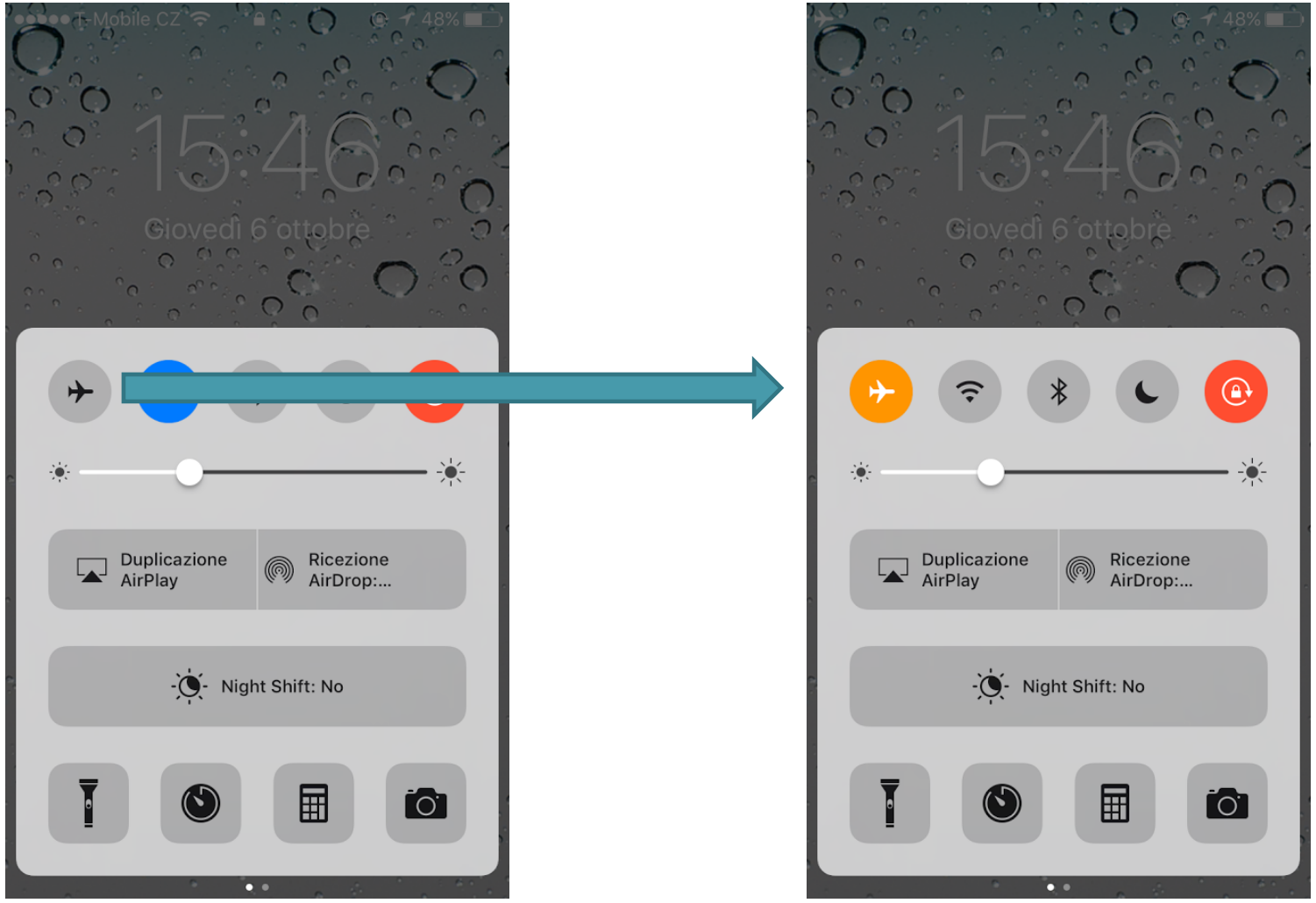
SEARCH AND SEIZURE OF iOS DEVICES

- Turned off device
  - **LEAVE IT OFF!**
- Turned on device (locked or unlocked)
  - **DON'T TURN IT OFF AND THINK!**

TURNED ON AND LOCKED

1. Activate **Airplane mode**
2. Connect to a **power source** (i.e. external battery)
3. Verify the **model**
4. Verify the **iOS version**

ACTIVATE AIRPLANE MODE
LOCKED DEVICE



IDENTIFY THE MODEL (1)

IDENTIFY THE MODEL (II) AND THE iOS VERSION

- **Libimobiledevice** (Linux/Mac)
  http://www.libimobiledevice.org/
- **iMobiledevice** (Windows)
  http://quamotion.mobi/iMobileDevice/
- **ideviceinfo -s**
- **They also work on locked devices!**

EXAMPLE

```
santoku@santoku:~$ ideviceinfo -s
BasebandCertId: 3840149528
BasebandKeyHashInformation:
 AKeyStatus: 2
 SKeyHash: u+/tcCwvaQ+1Y9t40I4yegCEmB28mALlaROhaIVGBWo=
 SKeyStatus: 0
BasebandSerialNumber: CKyShA==
BasebandVersion: 1.23.00
BoardId: 4
BuildVersion: 13D15
ChipID: 32771
DeviceClass: iPhone
DeviceColor: #272728
DeviceName: EpiPhone
DieID: 6299231647892006
HardwareModel: N71mAP
PartitionType:
ProductName: iPhone OS
ProductType: iPhone8,1
ProductVersion: 9.2.1
ProductionSOC: true
ProtocolVersion: 2
TelephonyCapability: true
UniqueChipID: 6299231647892006
UniqueDeviceID: 3bf682ebc55c5673d586e0273af0dfb72d1994a2
WiFiAddress: 1c:5c:f2:7f:7a:20
santoku@santoku:~$
```

REALITYNET
System Solutions

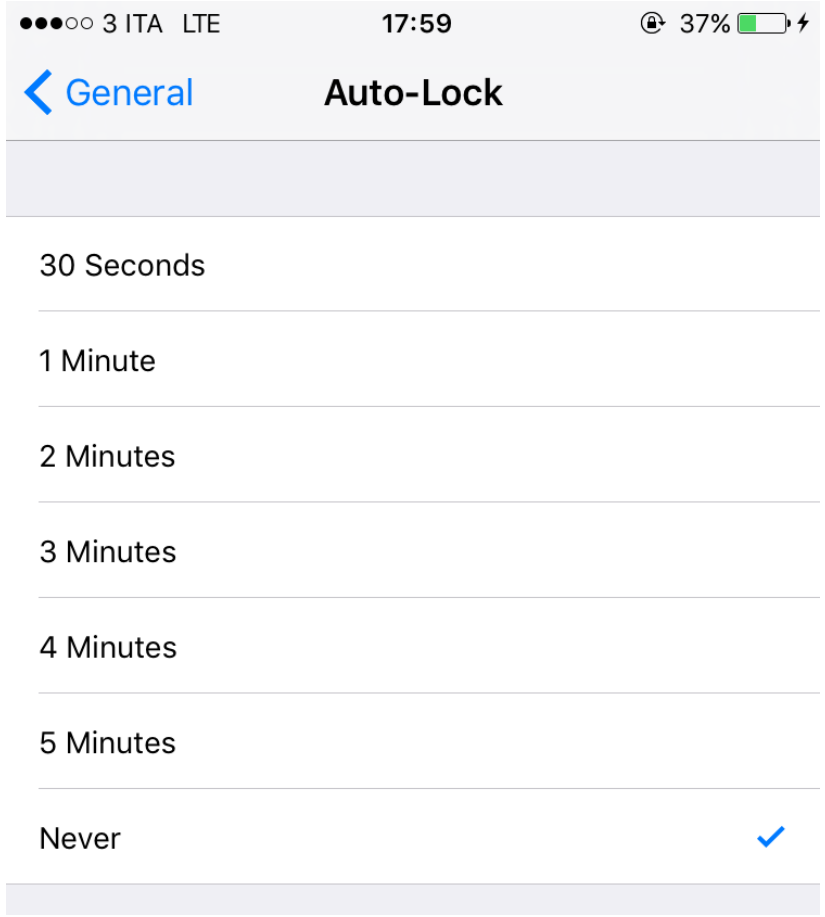
iPHONE MODEL CHART

Device name	Model number	Internal Name	Identifier	Year	Capacity (GB)
iPhone SE	A1662 – A1723 – A1724	N69AP	iPhone8,4	2016	16,64
iPhone 6s Plus	A1634 – A1687 – A1699 – A1690	N66AP	iPhone8,2	2015	16,64,128
iPhone 6s	A1633 – A1688 – A1700 – A1691	N71AP	iPhone8.1	2015	16,64,128
iPhone 6 Plus	A1522 – A1524 – A1593	N56AP	iPhone7,1	2014	16,64,128
iPhone 6	A1549 – A1586	N61AP	iPhone7,2	2014	16,64,128
iPhone 5S (CDMA)	A1457 – A1518 – A1528 – A1530	N53AP	iPhone6,2	2013	16, 32
iPhone 5S (GSM)	A1433 – A1533	N51AP	iPhone6,1	2013	16, 32, 64
iPhone 5C (CDMA)	A1507 – A1516 – A1526 – A1529	N49AP	iPhone5,4	2013	16, 32
iPhone 5C (GSM)	A1456 – A1532	N48AP	iPhone5,3	2013	16, 32
iPhone 5 rev.2	A1429 – A1442	N42AP	iPhone5,2	2012	16, 32, 64
iPhone 5	A1428	N41AP	iPhone5,1	2012	16, 32, 64
iPhone 4s (China)	A1431	N94AP	iPhone4,1	2011	8, 16, 32, 64
iPhone 4S	A1387			2011	8, 16, 32, 64
iPhone 4 - CDMA	A1349	N92AP	iPhone3,2	2011	8, 16, 32
iPhone 4 - GSM	A1332	N90AP	iPhone3,1	2010	8, 16, 32
iPhone 3GS (China)	A1325	N88AP	iPhone2,1	2009	8, 16, 32
iPhone 3GS	A1303			2009	8, 16, 32
iPhone 3G (China)	A1324	N82AP	iPhone1,2	2009	8, 16
iPhone 3G	A1241			2008	8, 16
iPhone 2G	A1203	M68AP	iPhone1,1	2007	4, 8, 16

TURNED ON AND UNLOCKED

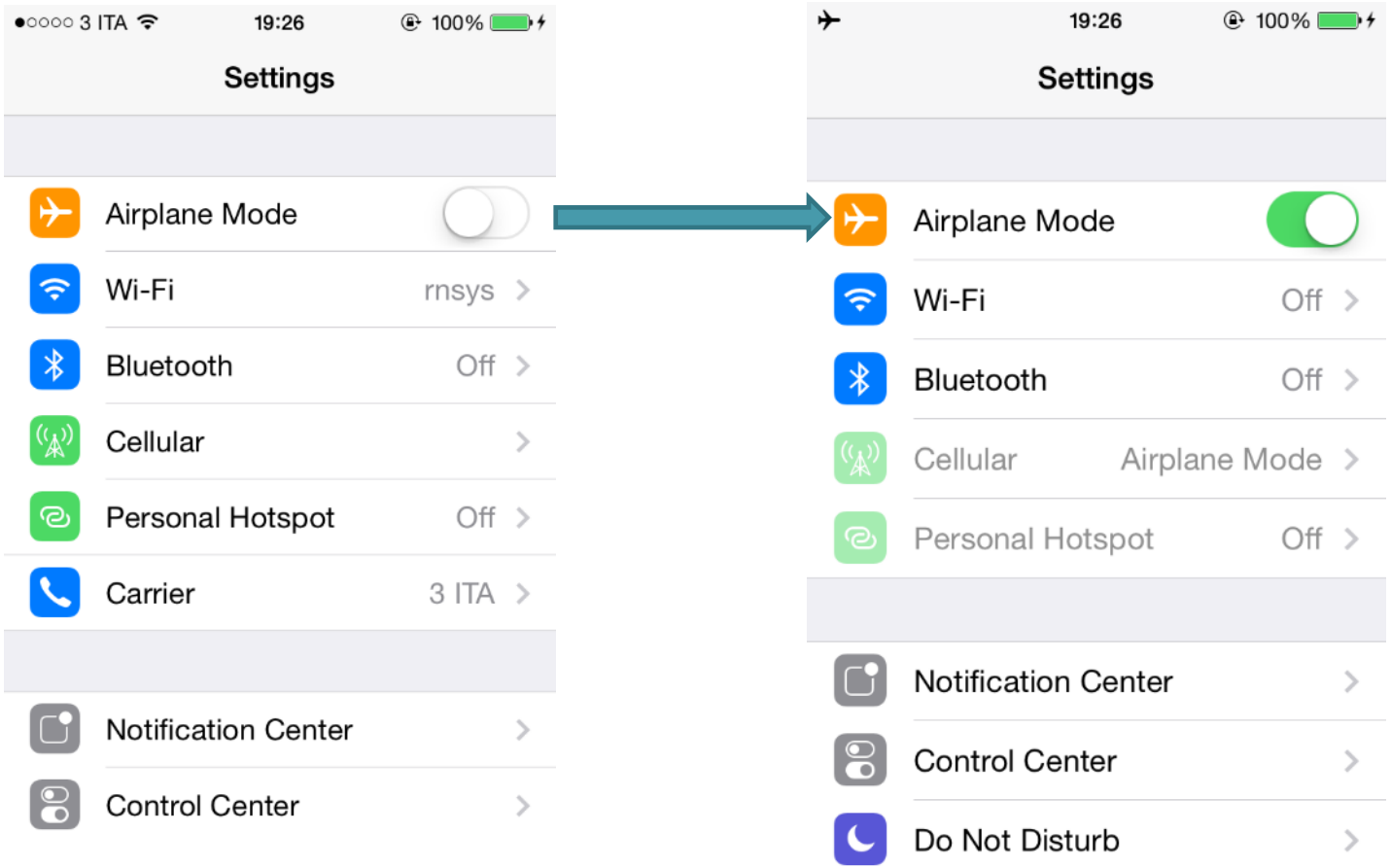
1. **Prevent the phone locking!**
   I. **Don’t press power button!**
   II. **Disable Auto-lock!**
2. **Verify if a lock code is set!**
3. Activate **Airplane mode**
4. Connect to a **power source** (i.e. external battery)
5. Identify the **model**
6. Identify the **iOS version**

PREVENT LOCK STATE! (DISABLE AUTO-LOCK)



REALITYNET
System Solutions

ACTIVATE AIRPLANE MODE UNLOCKED DEVICE



ACQUISITION TECHNIQUES

File System

- **iTunes Backup** Can be password protected!
- **Apple File Relay** Zdziarski, 2014 – Up to iOS 7
- **Apple File Conduit** Result depends on iOS version
- **iCloud** Already stored data or forced
- **Full file system** Possible only on jailbroken devices

Physical

- **Available up to iPhone 4**
- Possible on **jailbroken 32-bit devices**

iPHONE 4 AND BELOW

- A **physical acquisition is always possible**
- In case of **simple passcode** all data will be decrypted
- In case of **complex passcode** you will get in any case native applications data (i.e. address book, SMS, notes, video, images, etc.)

TURNED ON/OFF AND UNLOCKED

- Always possible doing some kind of **file system** acquisition
- The obtained data strongly depends on the iOS version
- Possible problems:
  - **Backup password**
  - **Managed devices → Connection to PC inhibited**

TURNED ON AND LOCKED

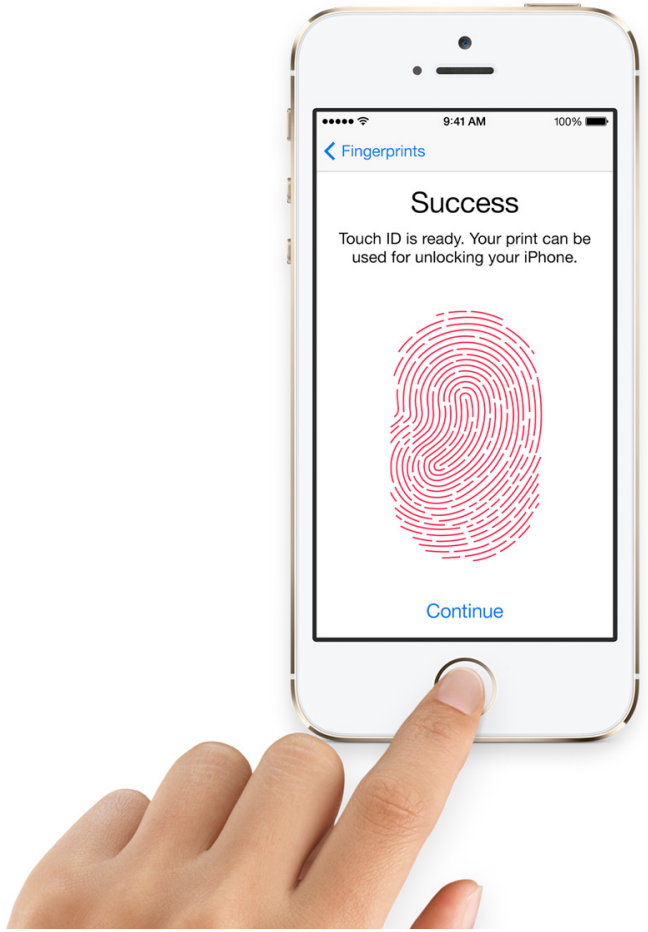
- Search for a **lockdown certificate** on a synced computer
- Unlock through **fingerprint**
- Try to **force an iCloud backup**
- Specific iOS version **vulnerability for bypassing passcode**

LOCKDOWN CERTIFICATE

- Stored in:
  - C:\Program Data\Apple\Lockdown — Win 7/8/10
  - C:\Users\[username]\AppData\roaming\Apple Computer\Lockdown — Vista
  - C:\Documents and Settings\[username]\Application Data\Apple Computer\Lockdown — XP
  - /private/var/db/lockdown — Mac OS X
- Certificate file name → **Device_UDID.plist**
- **The certificate can be extracted from the computer and used in another with some forensic tools or directly with iTunes**
- **Lockdown certificate stored on a computer is valid for 30 days**
- **Lockdown certificate can be used within 48 hours since last user unlocked with the passcode**

FINGEPRINT UNLOCK

- *To configure Touch ID, you must first set up a passcode. Touch ID is designed to minimize the input of your passcode; but your passcode will be needed for additional security validation:*
  - *After restarting your device*
  - *When more than 48 hours have elapsed from the last time you unlocked your device*
  - *To enter the Touch ID & Passcode setting*
- https://support.apple.com/en-us/HT204587



FORCE iCLOUD BACKUP

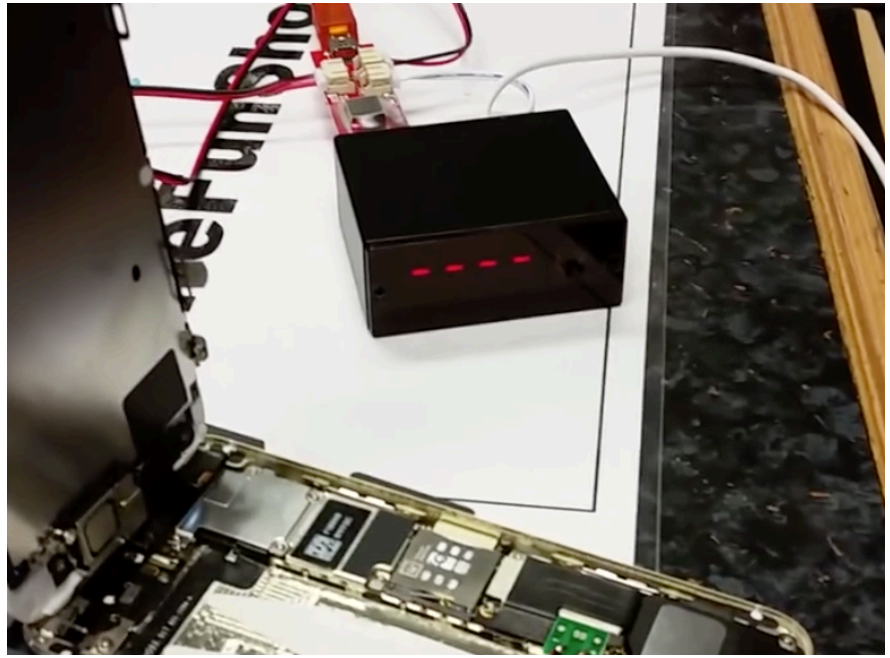
- **Be careful when using this option and try other methods first!**
  - **Possible overwriting of already existing backup**
  - **Risk of remote wiping**
- Follow this approach:
  - Bring the device close to a known Wi-Fi network
  - Connect to a power source
  - Wait a few hours
  - Request data from Apple or download it
    - **Legal authorization**
    - **Credentials or token is needed**

SPECIFIC iOS VULNERABILITY

- A comprehensive and continuously updated list is maintained at:
- **http://blog.dinosec.com/2014/09/bypassing-ios-lock-screens.html**
- Latest available for **iOS 9.3.1**
- CVE-2016-1852
  - *"Siri in Apple iOS before 9.3.2 does not block data detectors within results in the lock-screen state, which allows physically proximate attackers to obtain sensitive contact and photo information via unspecified vectors."*

TURNED OFF AND LOCKED

- Try to **use a lockdown certificate**
  - It works well on iOS 7 (AFR and AFC)
  - It can still get some data on iOS 8 (AFC)
  - Not useful on iOS 8/9/10
- Some specific **unlocking tools**
  - UFED User Lock Code Recovery Tool
  - IP-BOX
  - MFC Dongle
  - Xpin Clip



TURNED OFF AND LOCKED
CAIS (CELLEBRITE ADVANCED INVESTIGATIVE SERVICES)

Unmatched Device Access

Continuing its commitment to deliver digital forensic breakthroughs, Cellebrite now makes the world's first "decrypted physical extraction" capability a reality for key iPhone and Samsung Android devices. Notably, this new capability enables forensic examiners and investigators to access the full file system to recover downloaded emails, third-party application data, geolocation data, and system logs, without needing to jailbreak the device.

Cellebrite's exclusive unlocking and decrypted physical extraction capabilities support the following devices:

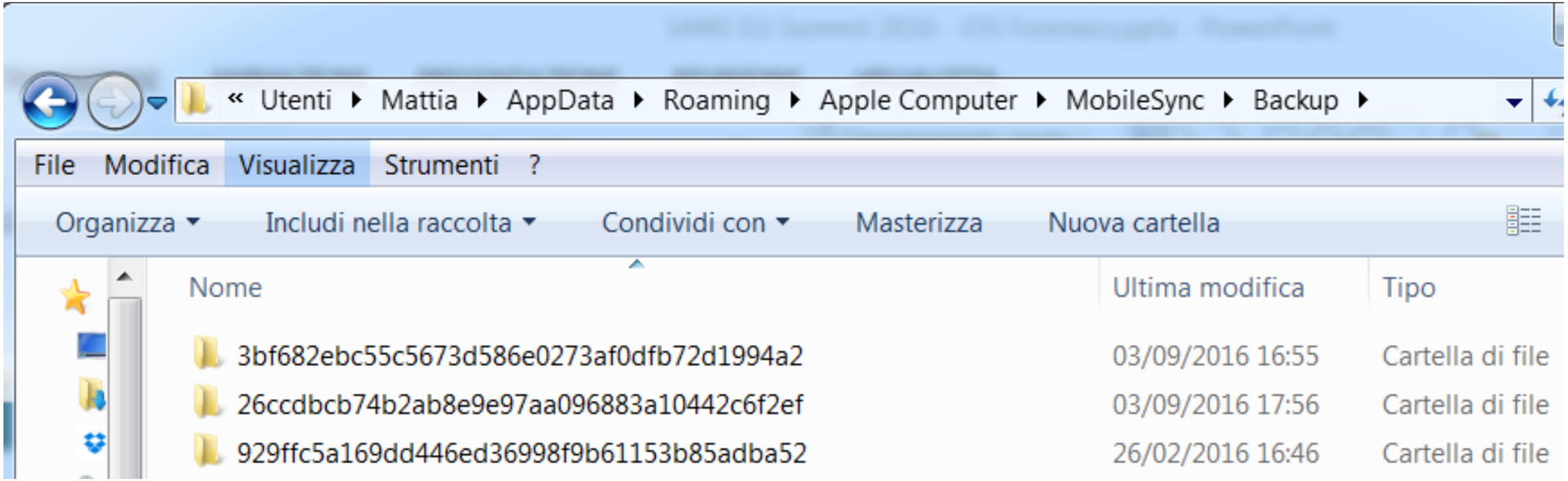
- iPhone 4S / 5 / 5c, iPad 2 / 3G / 4G, iPad mini 1G, and iPod touch 5G running iOS 8.x (8.0 / 8.0.1 / 8.0.2 / 8.1 / 8.1.1 / 8.1.2 / 8.1.3 / 8.2/ 8.3 / 8.4 / 8.4.1) or iOS 9.x (9.0 / 9.0.1 / 9.0.2 / 9.1 / 9.2 / 9.2.1 / 9.3 / 9.3.1 / 9.3.2)
- Samsung Galaxy S6, Galaxy Note 5 and Galaxy S7 running all Android versions up to and including Marshmallow 6.0.1

iOS ARTIFACTS: ALTERNATIVE OPTIONS

ALTERNATIVE OPTIONS

- **Local backup** stored on user's computer
- **Other data** stored on user's computer
- **Jailbreaking**
- **iCloud acquisition**
- **Experimental techniques** (chip-off)

BACKUP STORED ON THE USER'S COMPUTER



REALITYNET
System Solutions

ENCRYPTED BACKUP

iPhone 6s

Capacity: 55,53 GB
Phone Number: +39 334 2340899
Serial Number: F17QT811GRY9

iOS 9.3.5
Your iPhone software is up to date. iTunes will automatically check for an update again on 08/09/2016.

Check for Update | Restore iPhone...

Backups

Automatically Back Up

- iCloud
  Back up the most important data on your iPhone to iCloud.
- This computer
  A full backup of your iPhone will be stored on this computer.
  - Encrypt iPhone backup
    This will allowaccount passwords, Health, and HomeKit data to be backed up.
    Change Password...

Manually Back Up and Restore
Manually back up your iPhone to this computer or restore a backup stored on this computer.

Back Up Now | Restore Backup...

Latest Backup:
Today 22:13 to this computer

iOS 10 ENCRYPTED BACKUP WEAKNESS

iOS 10: Security Weakness Discovered, Backup Passwords Much Easier to Break

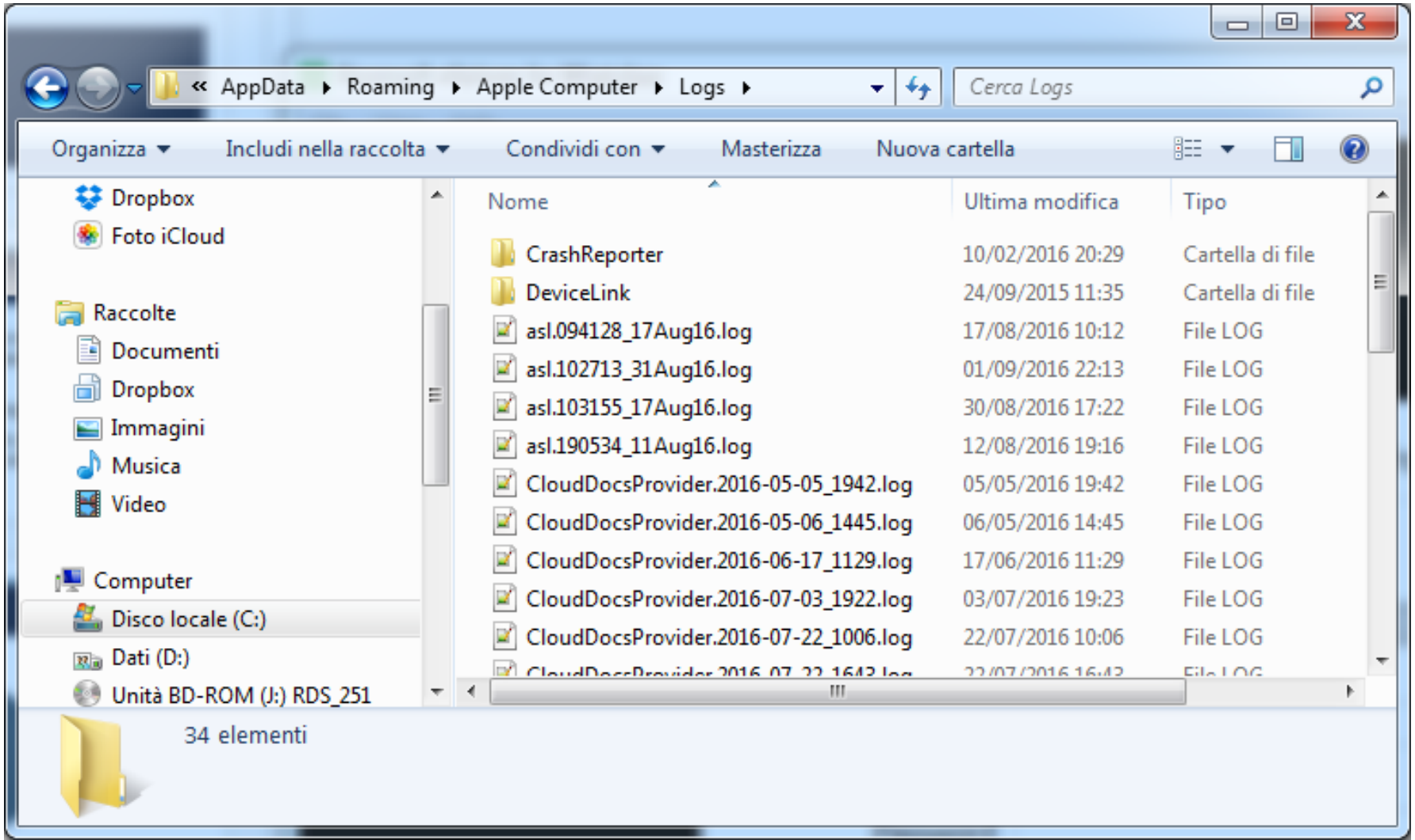
September 23rd, 2016 by Oleg Afonin

We discovered a major security flaw in the iOS 10 backup protection mechanism. This security flaw allowed us developing a new attack that is able to bypass certain security checks when enumerating passwords protecting local (iTunes) backups made by iOS 10 devices.

The impact of this security weakness is severe. An early CPU-only implementation of this attack (available in Elcomsoft Phone Breaker 6.10) gives a 40-times performance boost compared to a fully optimized GPU-assisted attack on iOS 9 backups.

http://blog.elcomsoft.com/2016/09/ios-10-security-weakness-discovered-backup-passwords-much-easier-to-break/

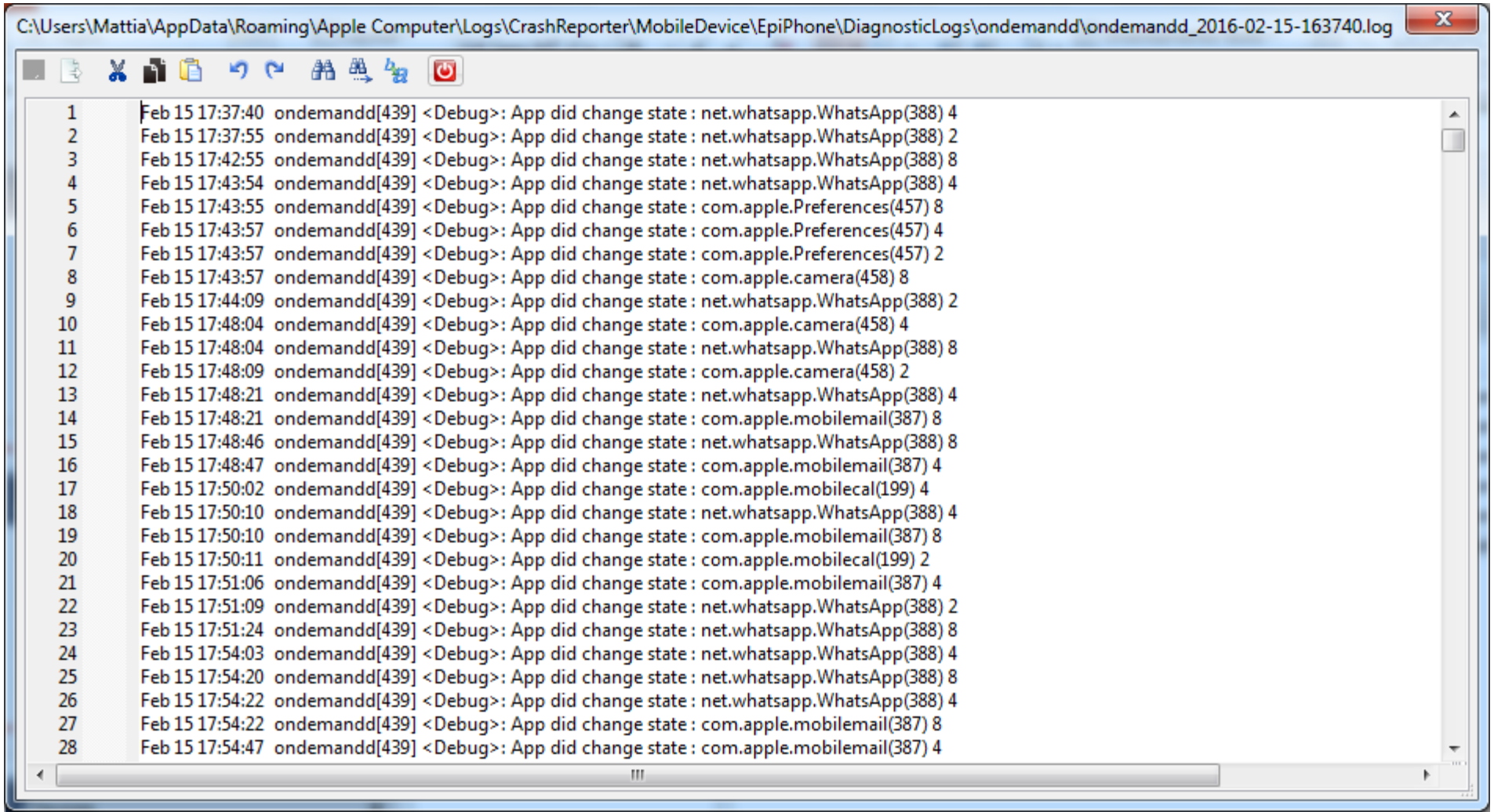
OTHER DATA STORED ON THE USER'S COMPUTER



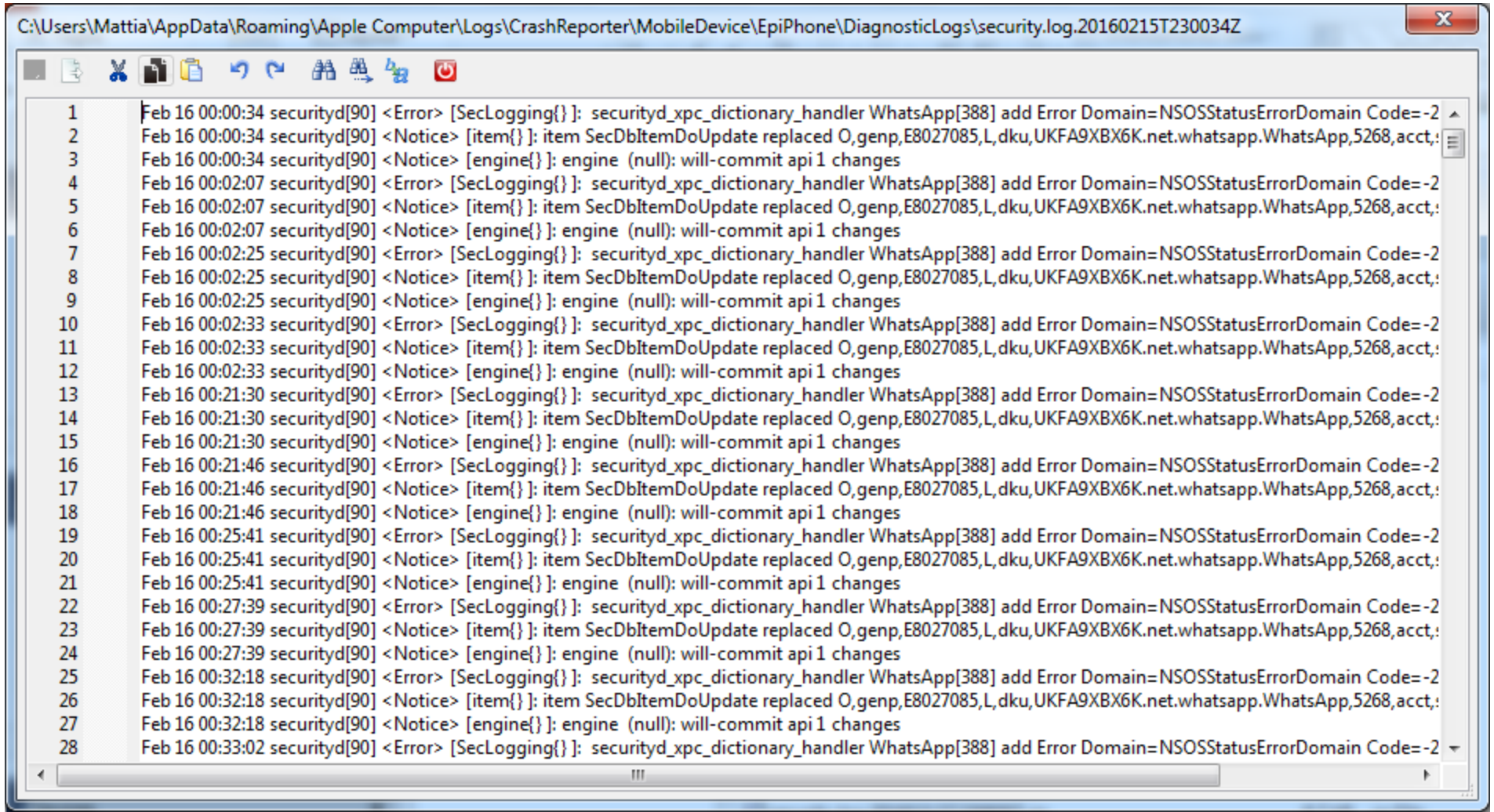
OTHER DATA STORED ON THE USER'S COMPUTER

- **Installed applications list and usage**
  - Various logs like **PowerLog, Security, OnDemand**
- **iTunes username**
  - **itunesstored.2.log** file
- **File name of e-mail attachments**
  - **MobileMail** logs
- **List of Wi-Fi networks and history of latest connections**
  - **Wi-Fi** logs

ONDEMAND LOG

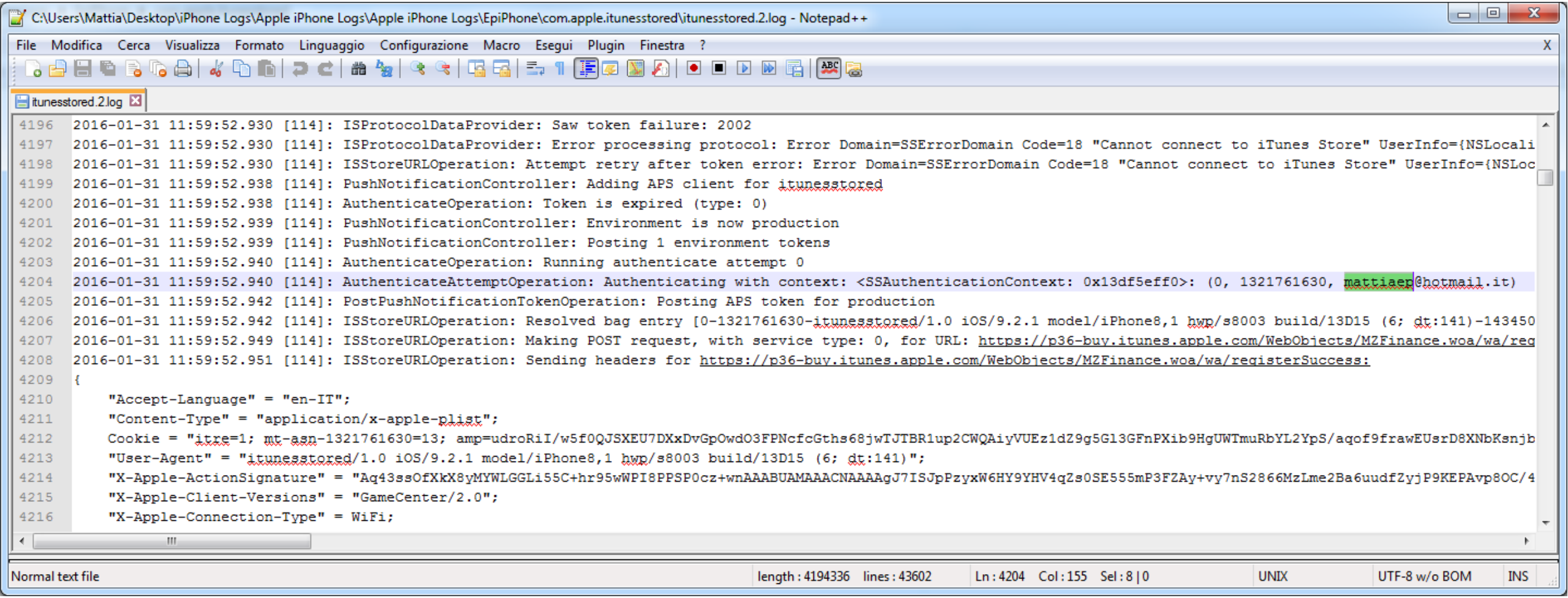


SECURITY LOG



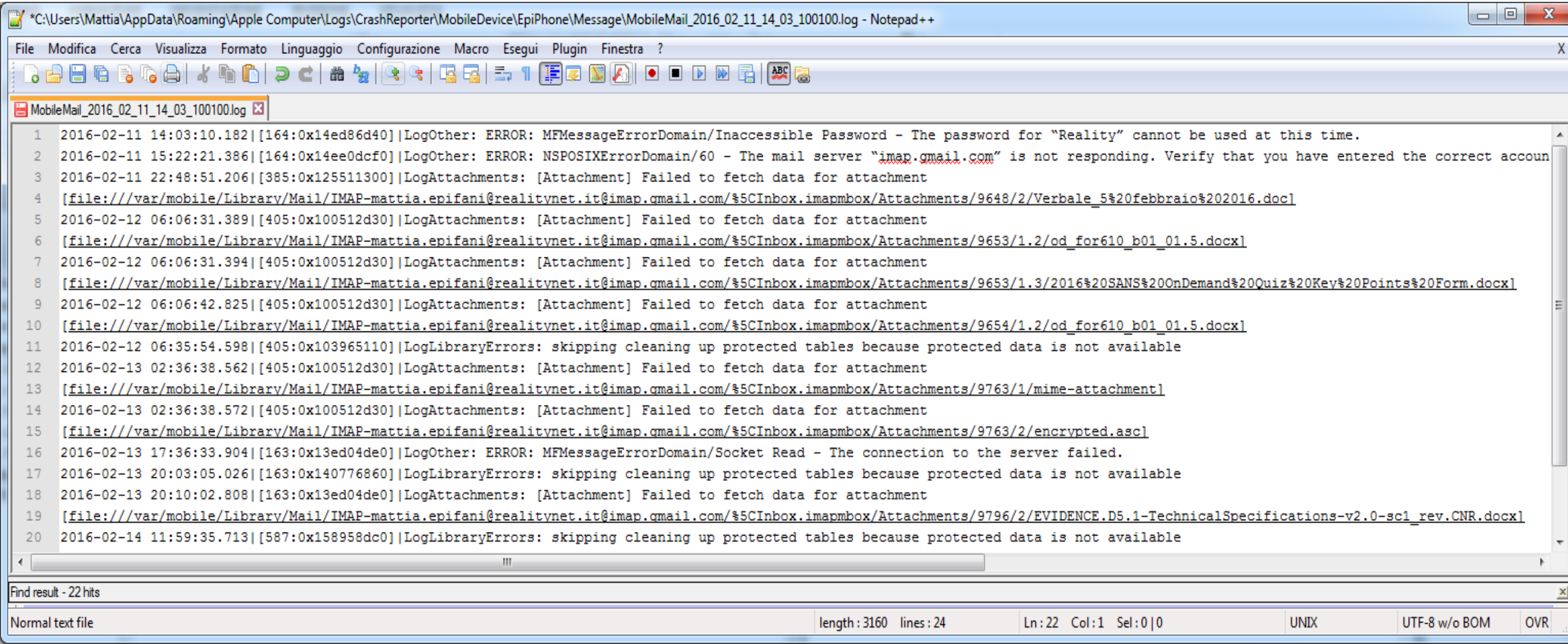
REALITYNET
System Solutions

ITUNESSTORED.2.LOG

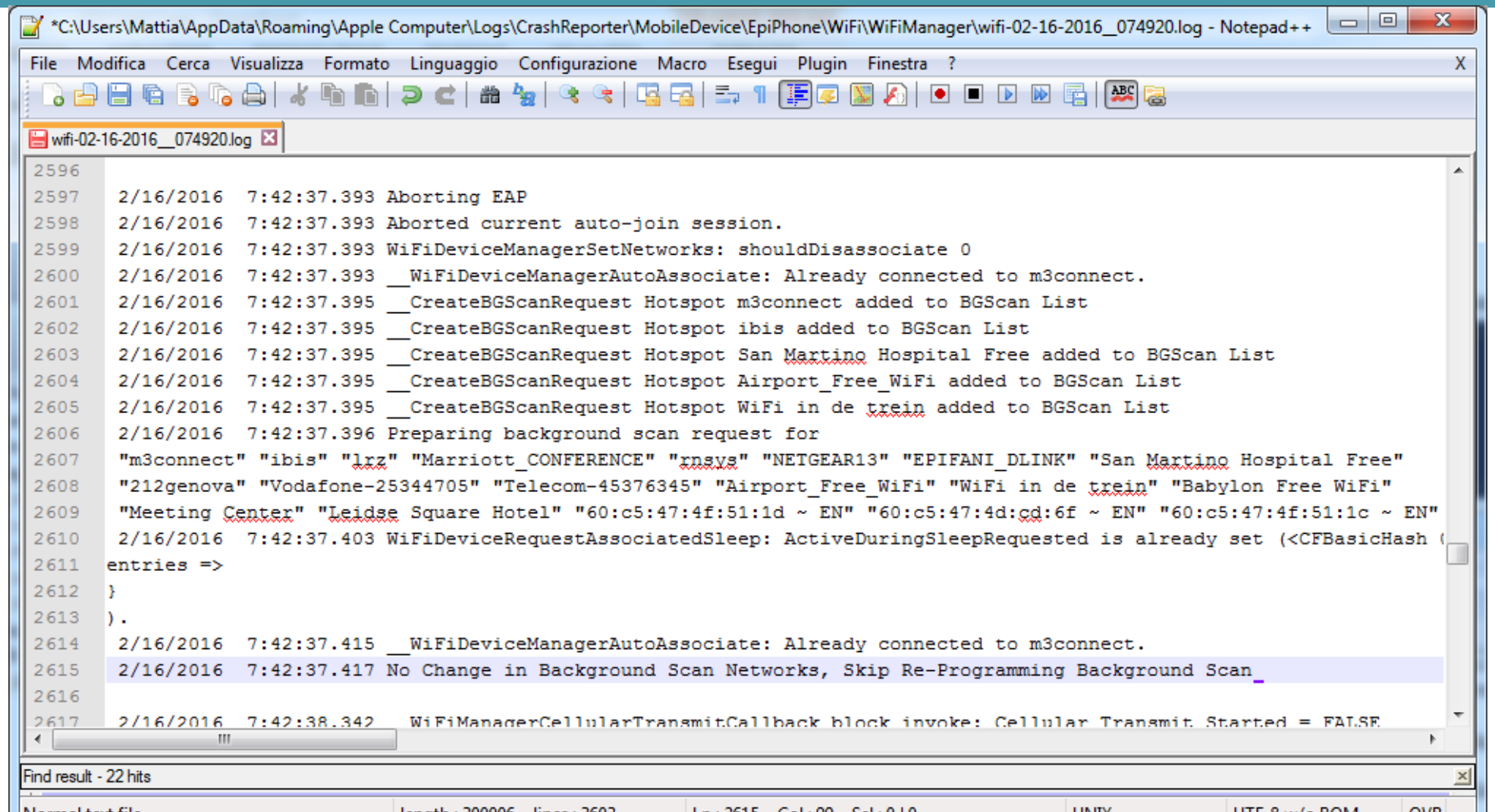


REALITYNET
System Solutions

MOBILEMAIL LOG



Wi-Fi LOG


*C:\Users\Mattia\AppData\Roaming\Apple Computer\Logs\CrashReporter\MobileDevice\EpiPhone\WiFi\WiFiManager\wifi-02-16-2016_074920.log - Notepad++
File Modifica Cerca Visualizza Formato Linguaggio Configurazione Macro Esegui Plugin Finestra ?
wifi-02-16-2016_074920.log
2596
2597 2/16/2016 7:42:37.393 Aborting EAP
2598 2/16/2016 7:42:37.393 Aborted current auto-join session.
2599 2/16/2016 7:42:37.393 WiFiDeviceManagerSetNetworks: shouldDisassociate 0
2600 2/16/2016 7:42:37.393 __WiFiDeviceManagerAutoAssociate: Already connected to m3connect.
2601 2/16/2016 7:42:37.395 __CreateBGScanRequest Hotspot m3connect added to BGScan List
2602 2/16/2016 7:42:37.395 __CreateBGScanRequest Hotspot ibis added to BGScan List
2603 2/16/2016 7:42:37.395 __CreateBGScanRequest Hotspot San Martino Hospital Free added to BGScan List
2604 2/16/2016 7:42:37.395 __CreateBGScanRequest Hotspot Airport_Free_WiFi added to BGScan List
2605 2/16/2016 7:42:37.395 __CreateBGScanRequest Hotspot WiFi in de trein added to BGScan List
2606 2/16/2016 7:42:37.396 Preparing background scan request for
2607 "m3connect" "ibis" "lrz" "Marriott_CONFERENCE" "rnsys" "NETGEAR13" "EPIFANI_DLINK" "San Martino Hospital Free"
2608 "212genova" "Vodafone-25344705" "Telecom-45376345" "Airport_Free_WiFi" "WiFi in de trein" "Babylon Free WiFi"
2609 "Meeting Center" "Leidse Square Hotel" "60:c5:47:4f:51:1d ~ EN" "60:c5:47:4d:cd:6f ~ EN" "60:c5:47:4f:51:1c ~ EN"
2610 2/16/2016 7:42:37.403 WiFiDeviceRequestAssociatedSleep: ActiveDuringSleepRequested is already set (<CFBasicHash
2611 entries =>
2612 }
2613).
2614 2/16/2016 7:42:37.415 __WiFiDeviceManagerAutoAssociate: Already connected to m3connect.
2615 2/16/2016 7:42:37.417 No Change in Background Scan Networks, Skip Re-Programming Background Scan
2616
2617 2/16/2016 7:42:38.342 __WiFiManagerCellularTransmitCallback_block_invoke: Cellular Transmit Started = FALSE
Find result - 22 hits

MOBILEMEACCOUNTS.PLIST

C:\Users\Mattia\AppData\Roaming\Apple Computer\Preferences\MobileMeAccounts.plist

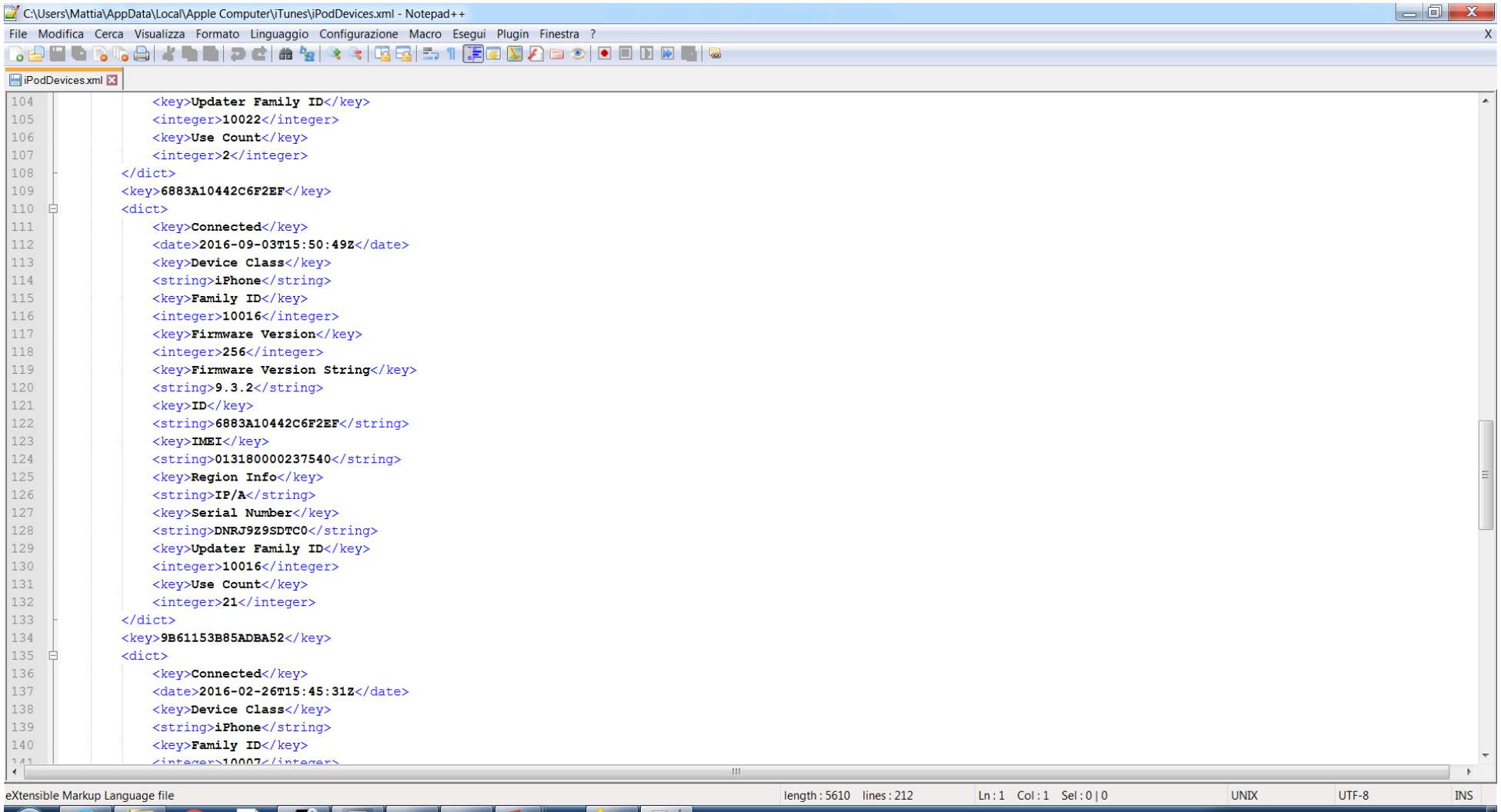
File Edit View Help

XML View | List View

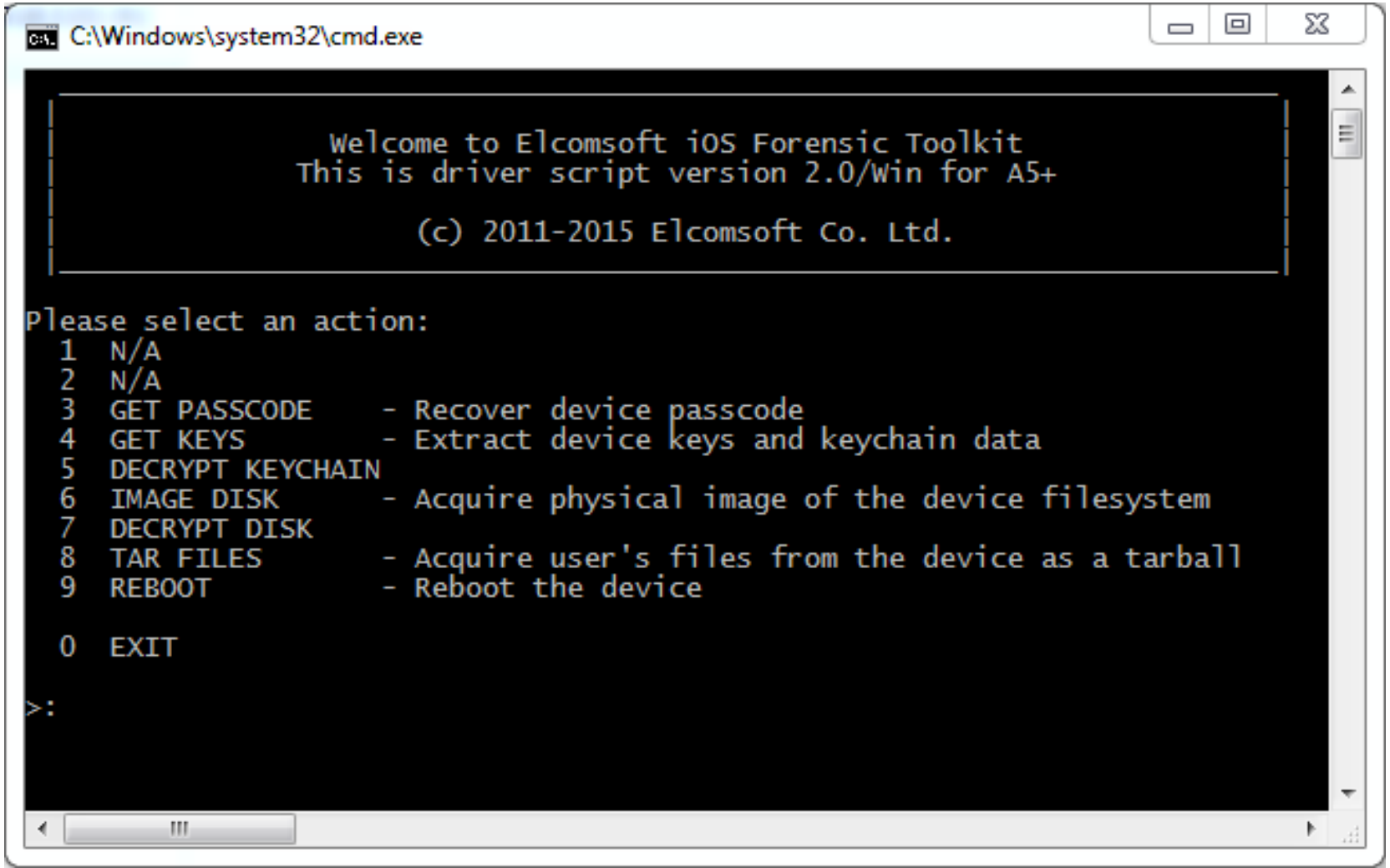
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
   <key>Accounts</key>
   <array>
      <dict>
         <key>AccountDSID</key>
         <string>461496998</string>
         <key>AccountDescription</key>
         <string>iCloud</string>
         <key>AccountID</key>
         <string>jailbreakingios@icloud.com</string>
         <key>DisplayName</key>
         <string>Jailbreaking Test</string>
         <key>IsPaidAccount</key>
         <false/>
         <key>LoggedIn</key>
         <true/>
         <key>Services</key>
         <array>
            <dict>
               <key>EmailAddress</key>
               <string>jailbreakingios@icloud.com</string>
               <key>Enabled</key>
               <true/>
               <key>FullUserName</key>
               <string>Jailbreaking Test</string>
```

REALITYNET
System Solutions

IPODDEVICES.XML

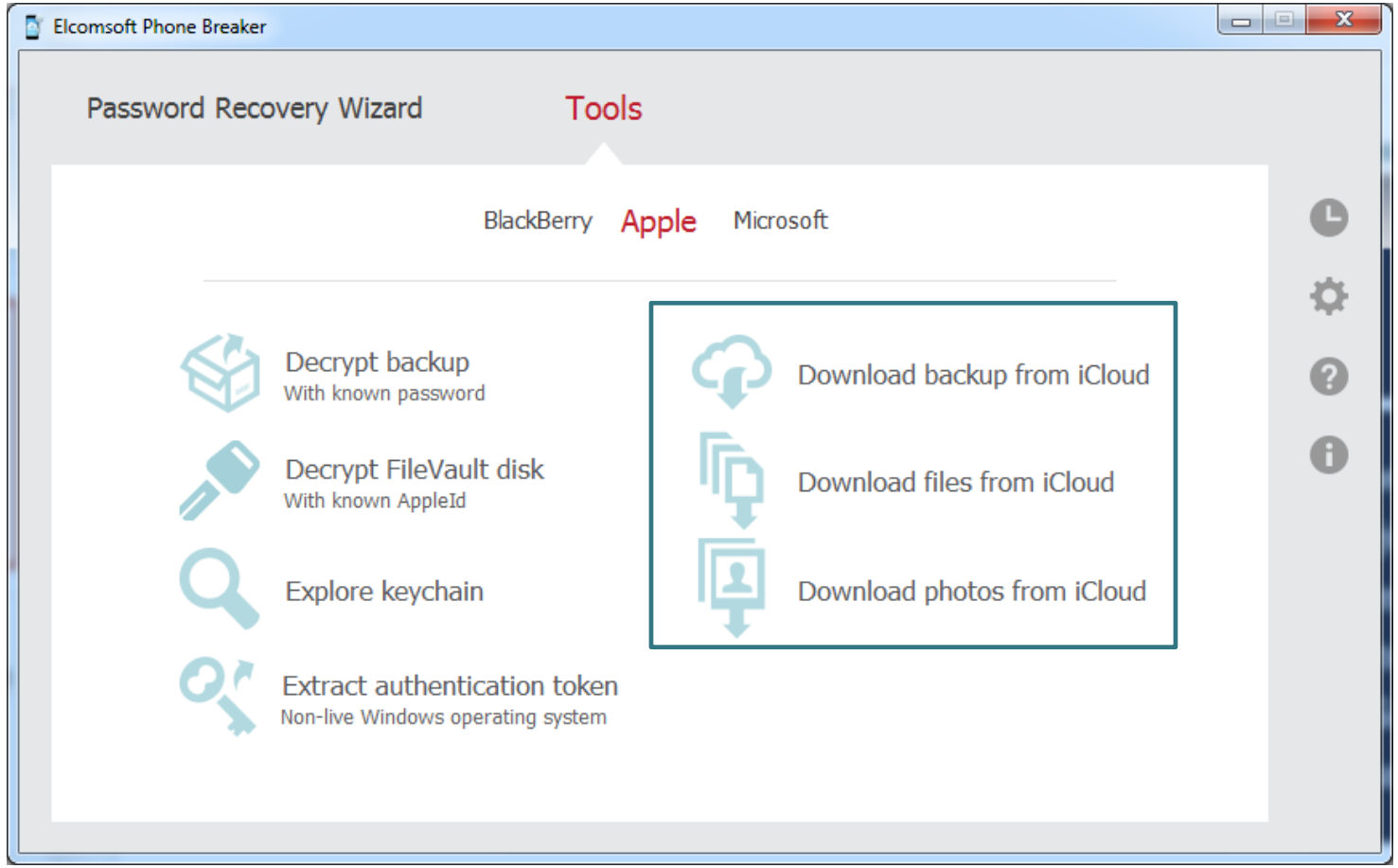


JAILBREAKING
HTTPS://WWW.THEIPHONEWIKI.COM/WIKI/JAILBREAK



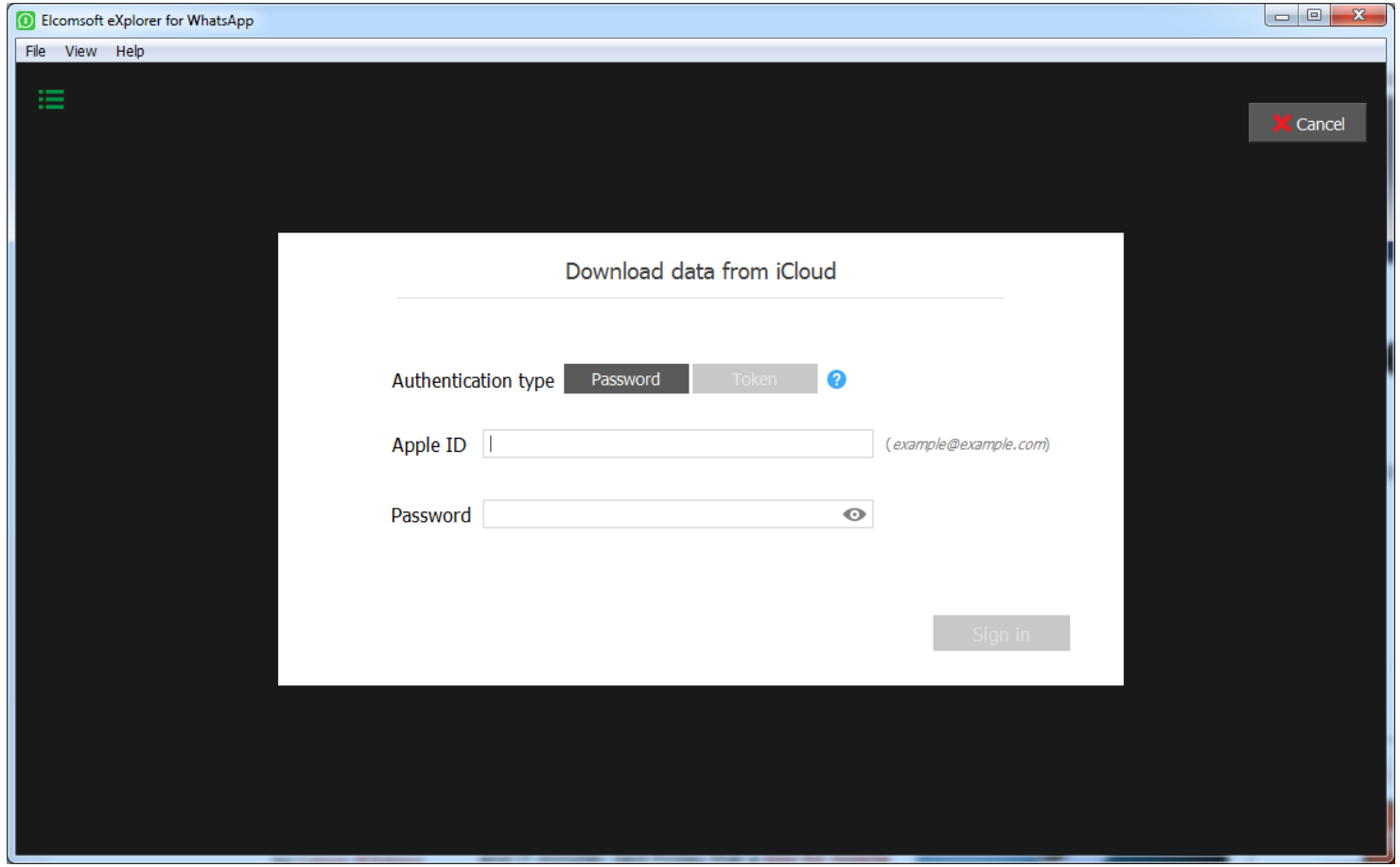
iCLOUD ACQUISITION

- You need **user credentials** or **token extracted from a computer (Windows/Mac) with iCloud Control Panel**
- You can obtain
  - iCloud Device Backup
  - iCloud Calendars
  - iCloud Contacts
  - Photo Streams
  - Email
  - Specific application data



iCLOUD ACQUISITION

- You need **user credentials** or **token extracted from a computer (Windows/Mac) with iCloud Control Panel**
- You can obtain
  - iCloud Device Backup
  - iCloud Calendars
  - iCloud Contacts
  - Photo Streams
  - Email
  - Specific application data

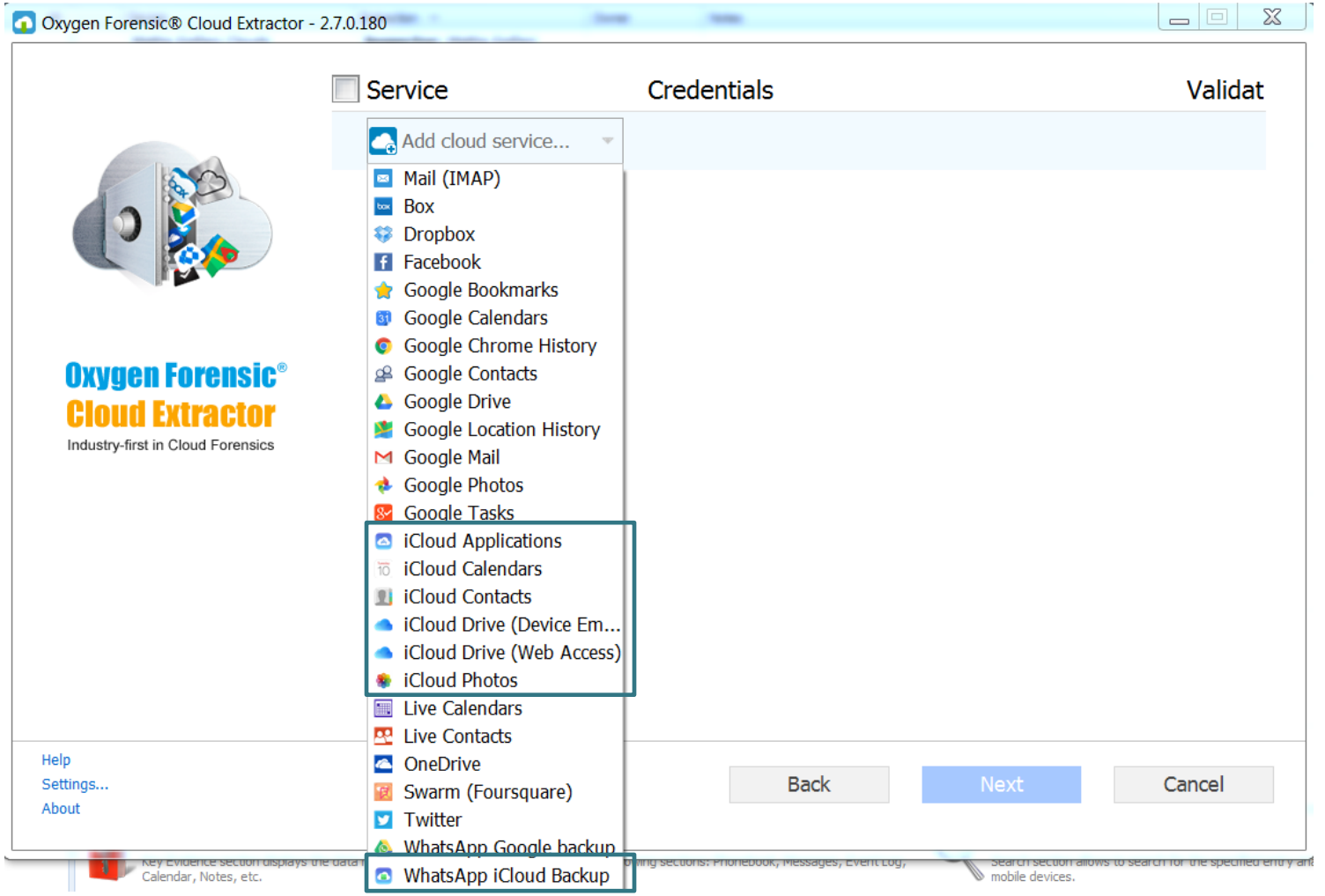


iCLOUD ACQUISITION

- You need **user credentials** or **token extracted from a computer (Windows/Mac) with iCloud Control Panel**
- You can obtain
  - iCloud Device Backup
  - iCloud Calendars
  - iCloud Contacts
  - Photo Streams
  - Email
  - Specific application data

iCLOUD ACQUISITION

- You need **user credentials** or **token extracted from a computer (Windows/Mac) with iCloud Control Panel**
- You can obtain
  - iCloud Device Backup
  - iCloud Calendars
  - iCloud Contacts
  - Photo Streams
  - Email
  - Specific application data



APPLE SUPPORT (iCLOUD DATA)
HTTP://IMAGES.APPLE.COM/PRIVACY/DOCS/LEGAL-PROCESS-GUIDELINES-US.PDF

- You can request:
  - Subscriber information
  - Mail logs
  - Email content
  - Other iCloud Content (iOS Device Backups, Photo Stream, Docs, Contacts, Calendars, Bookmarks)
  - Find My iPhone
  - Game Center
  - iOS Device Activation
  - Sign-on logs
  - My Apple ID and iForgot logs
  - FaceTime logs

i. Subscriber Information

When a customer sets up an iCloud account, basic subscriber information such as name, physical address, email address, and telephone number may be provided to Apple. Additionally, information regarding iCloud feature connections may also be available. iCloud subscriber information and connection logs with IP addresses can be obtained with a subpoena or greater legal process. Connection logs are retained up to 30 days.

EXPERIMENTAL TECHNIQUES - CHIP OFF

- Recently published research by Sergei Skorobogatov
- **The bumpy road towards iPhone 5C NAND mirroring**
  - http://www.cl.cam.ac.uk/~sps32/5c_proj.html
  - https://arxiv.org/pdf/1609.04327v1.pdf
  - https://www.youtube.com/watch?v=tM66GWrwbsY

iOS ACQUISITION TOOLS

Forensic Tools

- Cellebrite Physical Analyzer
- Oxygen Forensic
- Elcomsoft Phone Breaker
- Elcomsoft iOS Forensic Toolkit
- Magnet Acquire
- XRY
- MPE+
- Paraben Device Seizure

Other tools

- iTunes
- Libimobiledevice
- iMobiledevice
- iFunBox
- iTools
- iExplorer

iOS APPLICATION ANALYSIS

HOW AND WHERE DATA IS STORED

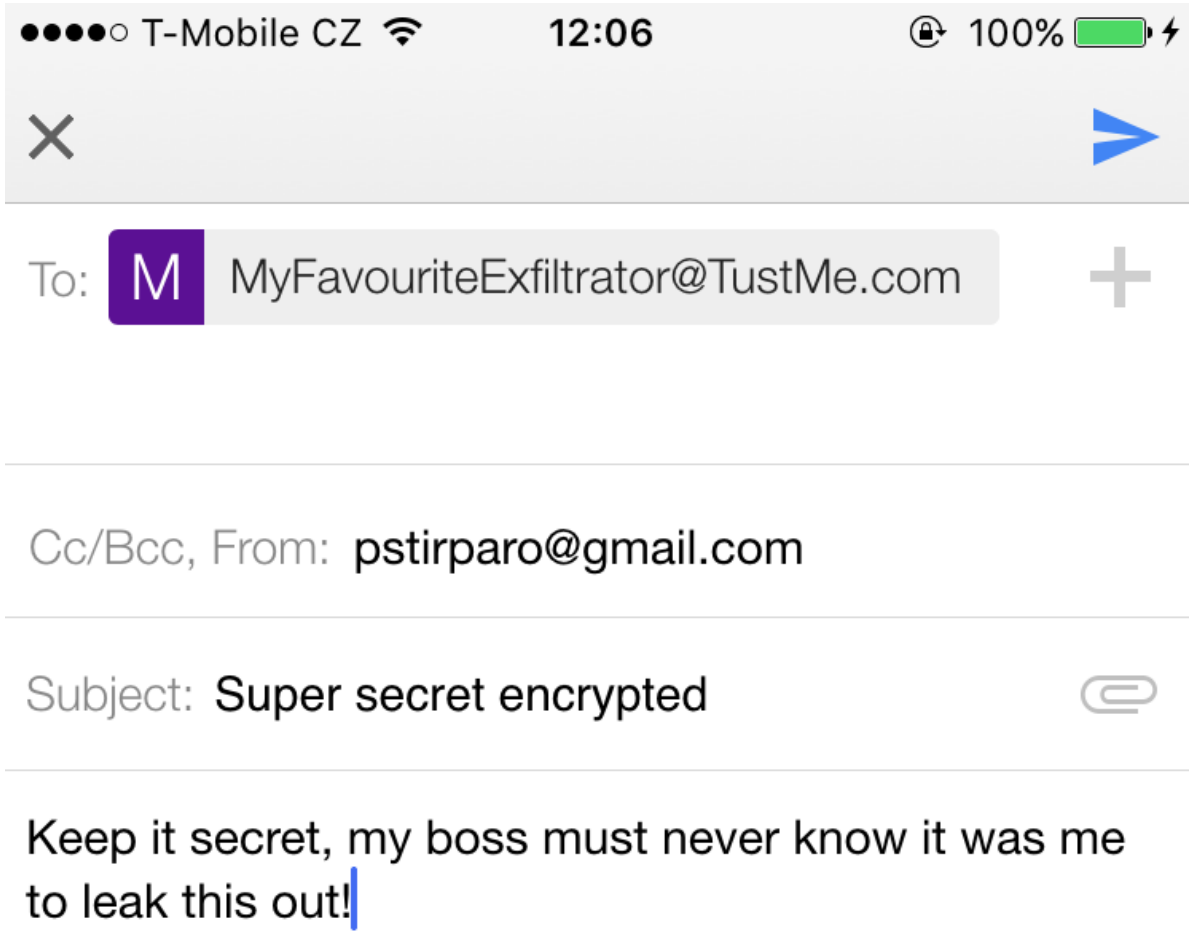
Starting from iOS8, application data have been separated from their bundles and current directory structure is the following

- **/private/var/mobile/Containers/Bundle/Application/<UUID>/**: This path is the actual path where the application bundle is stored.
- **/private/var/mobile/Containers/Data/Application/<UUID>/**: This path is the actual path where most of the application data is stored.
- **/private/var/mobile/Containers/Shared/AppGroup/<UUID>/**: As the name of the folder suggests, this path is the path where applications can store data with the aim of sharing it with other apps or extensions.

File formats are the usual plist files and SQLite databases

DO NOT FORGET THE FADE-OUT EFFECT

- Every time a user presses the Home button or receives a call while using an application, iOS will make a "snapshots" of the current screen in order to be able to do the fade-out effect transition between the two screens.
- **/private/var/mobile/Library/Caches/Snapshots/** for the pre-installed Apple applications;
- **/private/var/mobile/Containers/Data/Applications/<UUID>/Library/Caches/Snapshots/**, for each third-party application



DO NOT FORGET THE FADE-OUT EFFECT

"SECURE" MESSAGING IN iOS

- WhatsApp, Telegram and Signal among the most widespread applications for instant messaging.
- All of them claim "secure" messaging to a certain extent.
- All of them have end-to-end encryption (data-in-transit), but we will focus on the artifacts left on the device (data-at-rest).

WHATSAPP

- Data is stored in the Shared directory instead of the application data directory
  `/private/var/mobile/Containers/Shared/AppGroup/332A098D-368C-4378-A503-91BF33284D4B`
  `|-- Axolotl.sqlite`
  `|-- ChatSearch.sqlite`
  `|-- **ChatStorage.sqlite**`
  `|-- Contacts.sqlite`
- Some of the tables of interest are:
  - *ZWACHATSESSION*, *ZWAGROUPMEMBER*, *ZWAGROUPINFO* and *ZWAMEDIAITEM*, which stores references to the multimedia files exchanged, indication of the users involved, timestamps, the path where the file has been stored, etc.

WHATSAPP

- *ZWAMESSAGE* contains, among others, the messages exchanged, their timestamp, the name of the user involved in the chat.

Table: ZWAMESSAGE

New Record | Delete Record

	ZMESSAGEDATE	ZSENTDATE	ZFROMJID	ZPUSHNAME	ZSTANZAID	ZTEXT	ZTOJID
1	429476024.647006				140	Pro	39329
2	429476038.539024				140	Pro	39349
3	429476152		39329	Ric ssi	140	Nor	
4	429477100		39329	Ric ssi	140	Rag poi	
5	429575262.41247				140	Pro	39349
6	429716264.971918				140	Pro	39349
7	429716349.770499				140	Arip	39349
8	429716405.619276				140	Giu lta	39349
9	432568980.222825	432568980			141	Hi t mbe	44797
10	432569000.474907	432569000			141	Pas	44797
11	433081774		44797		141	447 het	39366
12	433081774		44797		141	DFI	39366
13	433082195		44797	An	141	Hi (	
14	433081856		44797	Ga ce	141	Just gro	
15	433082599		44797	Ga ce	141	Hi A	
16	433350110.17942	433350110			141	Hi g	44797
17	433350148		44797	Ga ce	141	Hey	
18	433351717.501071	433351717			141	Can	44797
19	433352062.324283	433352062			141	Btw unc	44797
20	433444570		39349	sil etti	141	Cia	

WHATSAPP: WHAT ABOUT DELETED DATA?

- As also recently mentioned by J. Zdziarski on his blog [1], an interesting "feature" of WhatsApp is that deleted chats are not actually deleted form the database.
- This because when a SQLite record is being deleted, for performance reasons it is not actually wiped/purged from the database immediately, but marked as free and eventually overwritten later on when that storage space is needed.
- With tools like SQLite-parser [2][3], you can quickly carve out deleted record from WhatsApp chat database.
- However, you will find this "feature" in most applications using SQLite storage databases, not just WhatsApp… keep that in mind.

[1] – "WhatsApp Forensic Artifacts: Chats Aren't Being Deleted", http://www.zdziarski.com/blog/?p=6143
[2] – "Python Parser to Recover Deleted SQLite Database Data", http://az4n6.blogspot.ch/2013/11/python-parser-to-recover-deleted-sqlite.html
[3] – "SQLite-parser", https://github.com/mdegrazia/SQLite-Deleted-Records-Parser

TELEGRAM

- Like WhatsApp, also Telegram stores many of its data in the *Shared* directory.
- The **tgdata.db** database, under the *Documents* folder, contains all information about contacts, conversations, exchanged files, etc.:
  - `messages_v29` contains the list of all messages exchanged
  - `convesations_v29` contains the list of active conversations as showed in the "Chats" screen of the app
  - `encrypted_cids_v29` contains the conversation ids of the secret chats.

TELEGRAM

- As expected, also with Telegram is possible to carve out deleted records from SQLite database… but there is one more "feature".
- Telegram **messages from secret chats are stored in clear** in the `messages_v29` table, like all the other messages.
- On the other hand **we will not find the screen snapshot**, as apparently Telegram properly clears the screen when the fade-out event happens.

SIGNAL

- Less popular than the previous two, but still important to know.
- It delivers what promises: its database **/Document/Signal.sqlite**, containing all its data, **is fully encrypted**. However, two things that are in clear:
  - The attachments exchanged are stored in clear in the /Document/Attachments/ folder.
  - Screen Snapshots can be retrieved as well. Signal has an option "**Enable Screen Security**" that would prevent this, but for some reason is not set by default.

iOS MESSAGING RECAP

	WhatsApp	Telegram	Signal
Standard message content in clear?			
“Secret chat” message content in clear?	n/a		
Sender/recipient information?			
Timestamps?			
SQLite carving of deleted records?			
Snapshot?			

TRACKING DEVICE USAGE THROUGH APPLICATION ANALYSIS

- Sarah Edwards has made an extensive research work on artifacts that track the usage of the device.
- By linking applications, data and network usage, health information (e.g. workouts), timestamps and geolocation data, it is indeed possible to understand what a given user under investigation was doing and where, at a given point in time.

[1] – "The iOS of Sauron: How iOS Tracks Everything You Do", https://www.mac4n6.com/resources/

TRACKING DEVICE USAGE THROUGH APPLICATION ANALYSIS

- CoreDuet: `/private/var/mobile/Library/CoreDuet/`
  - coreduetd.db (31 tables)
  - coreduetdClassA.db (31 tables)
  - coreduetdClassD.db (31 tables)
  - Knowledge/knowledgeC.db (5 tables)
  - People/interactionC.db (9 tables)
- Battery Life (PowerLog): `/private/var/mobile/Library/BatteryLife/`
  - CurrentPowerlog.PLSQL (257 tables)
  - Archives/powerlog_YYYY-MM-DD_XXXXXXXX.PLQSQL.gz (Previous ~5 Days)
- Health: `/private/var/mobile/Library/Health/`
  - healthdb.sqlite (11 tables)
  - healthdb_secure.sqlite (16 tables)
- Aggregate Dictionary: `/private/var/mobile/Library/AggregateDictionary/`
  - ADDataStore.sqlitedb (4 tables)
- networkd: `/private/var/networkd/`
  - netusage.sqlite (13 tables)
- routined: `/private/var/mobile/Library/Caches/com.apple.routined/`
  - cache_encryptedB.db (5 tables)
  - StateModel1.archive
  - StateModel2.archive
- locationd: `/private/var/root/Library/Caches/locationd/`
  - cache_encryptedA.db (79 tables)
  - lockCache_encryptedA.db (51 tables)
  - cache_encryptedB.db (167 tables)
  - cache_encryptedC.db (9 tables)

[1] – "The iOS of Sauron: How iOS Tracks Everything You Do", https://www.mac4n6.com/resources/

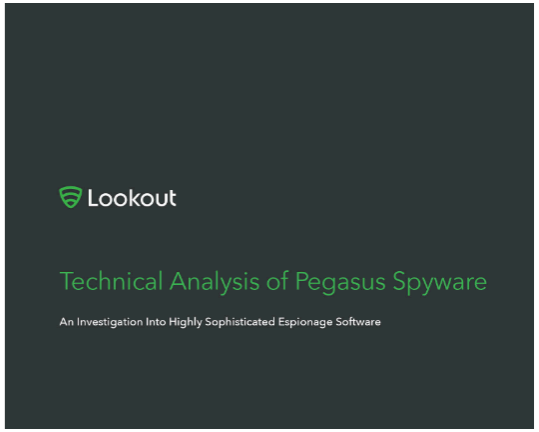
iOS MALWARE

August 25, 2016

Sophisticated, persistent mobile attack against high-value targets on iOS

By Lookout and Citizen Lab 7 Comments

Persistent, enterprise-class spyware is an underestimated problem on mobile devices. However, targeted attack scenarios against high-value mobile users are a real threat.



Citizen Lab (Munk School of Global Affairs, University of Toronto) and Lookout have uncovered an active threat using three critical

https://blog.lookout.com/blog/2016/08/25/trident-pegasus/

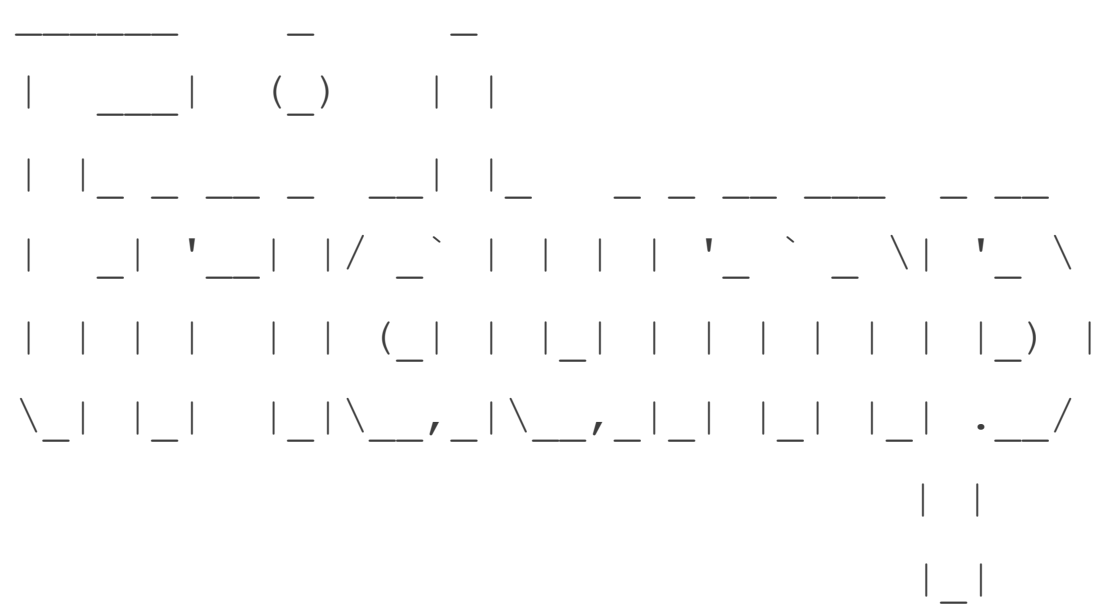
iOS MEMORY ANALYSIS… WHAT'S THAT?

- Memory analysis in the mobile domain not much developed, particularly in iOS
- Although rarely usable, may be of help in case of a running malware
- Current tools available consists in utilities/PoC
- *Frida* is a dynamic instrumentation framework that allows to inject scripts into processes to execute custom debugging logic.
- **Fridump** is a memory dumping tool built on top of Frida

```
$ frida-ps -U
PID  Name
---- ---------------
1744 Cydia
 137 Mail
1738 Settings
1808 Skype
  78 BTServer
1792 CacheDeleteAppCo ...
```

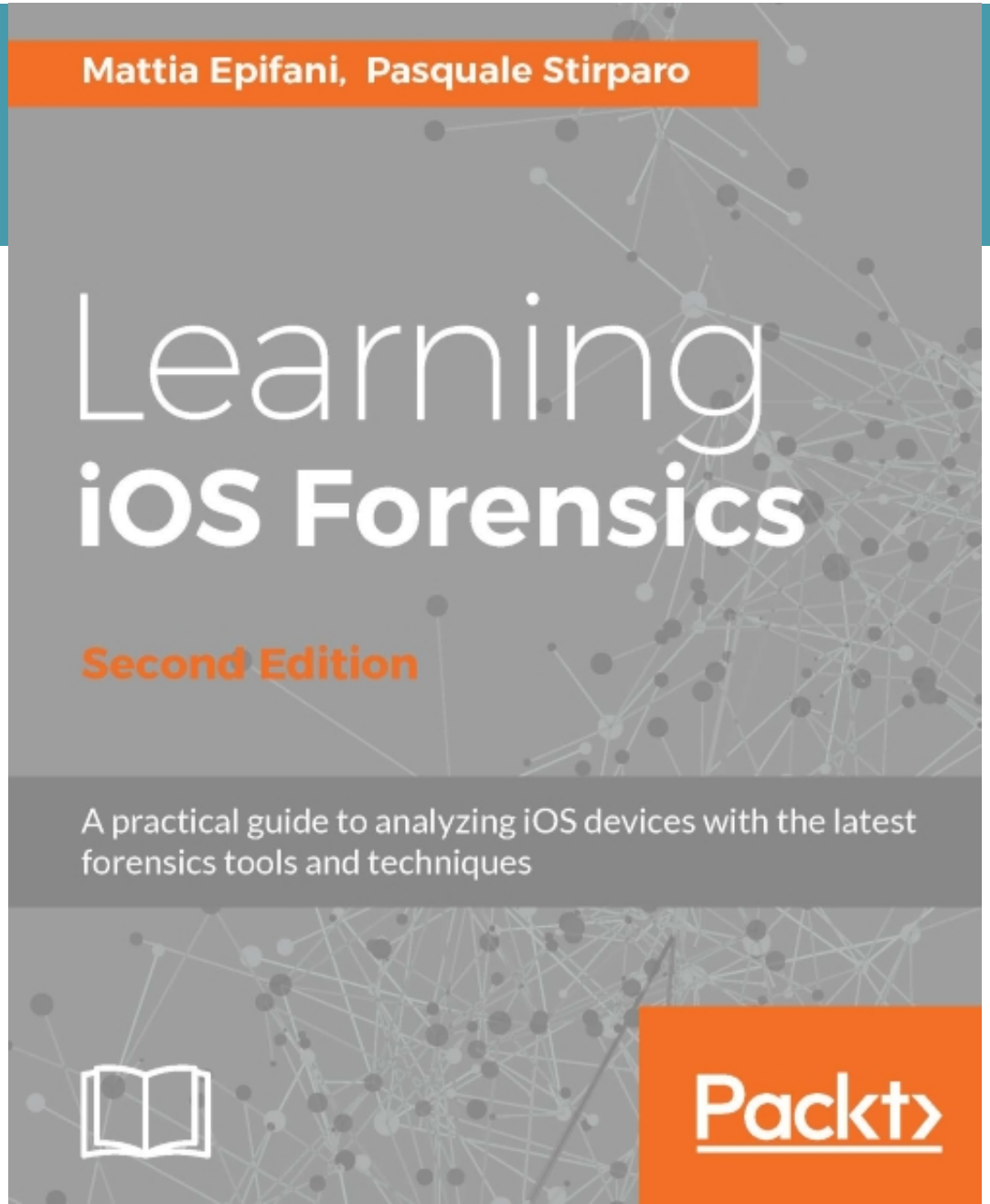
iOS MEMORY ANALYSIS… WHAT'S THAT?

```
$ python fridump.py -u -s --max-size 1048576 Skype
```



```
Current Directory: /Projects/iosmem/fridump
Output directory is set to: /Projects/iosmem/dump
Creating directory...
Starting Memory dump...
Progress: [##########################################----]88.42%
```

LEARNING iOS FORENSICS SECOND EDITION



https://www.packtpub.com/networking-and-servers/learning-ios-forensics-second-edition

Q&A?

Mattia Epifani

- Digital Forensics Analyst
- CEO @ REALITY NET – System Solutions
  Genoa, Italy
- GCFA, GMOB, GNFA, GREM, GCWN


mattia.epifani@realitynet.it


@mattiaep


http://www.linkedin.com/in/mattiaepifani


http://www.realitynet.it


http://blog.digital-forensics.it

Pasquale Stirparo, Ph.D.

- Threat Intelligence Analyst and Incident Responder @ UBS
- Incident Handler @ SANS ISC, Advisor @ Europol EC3
- GCFA, GREM, OPST, OWSE, ECCE


pstirparo@gmail.com


@pstirparo


http://www.linkedin.com/in/pasqualestirparo


https://isc.sans.edu