SANS

www.sans.org

FORENSICS 518

MAC FORENSIC ANALYSIS

518.5

iOS Forensics

The right security training for your staff, at the right time, in the right location.

Copyright © 2015, The SANS Institute. All rights reserved. The entire contents of this publication are the property of the SANS Institute.

IMPORTANT-READ CAREFULLY:

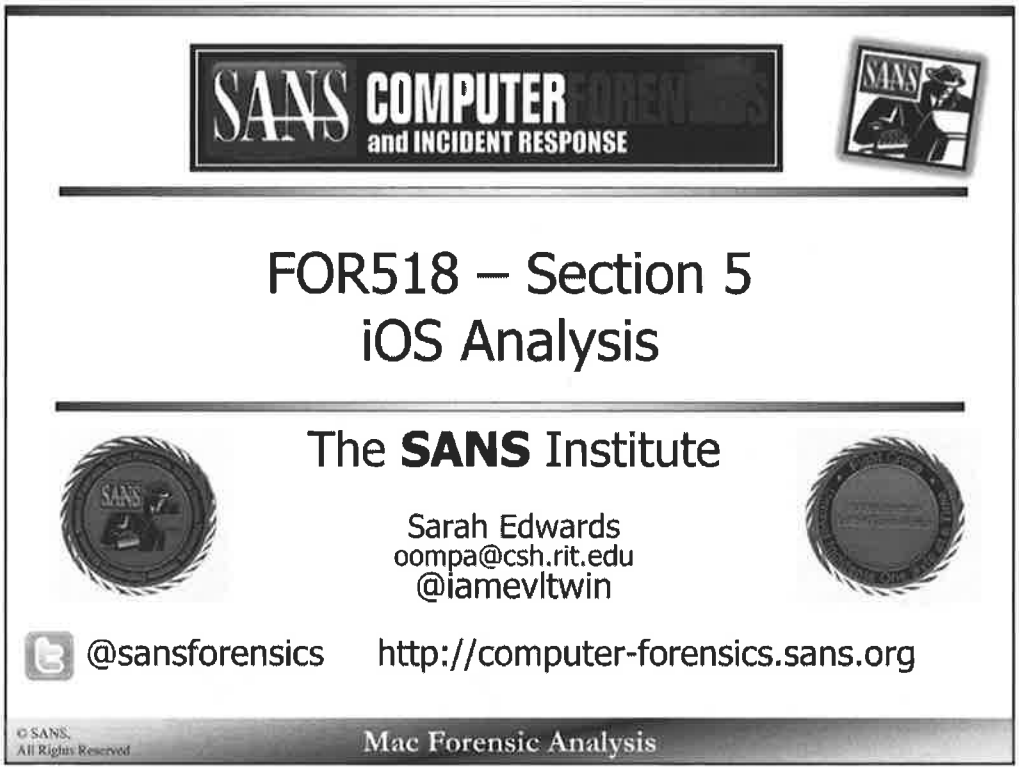
This Courseware License Agreement ("CLA") is a legal agreement between you (either an individual or a single entity; henceforth User) and the SANS Institute for the personal, non-transferable use of this courseware. User agrees that the CLA is the complete and exclusive statement of agreement between The SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA. If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this courseware. BY ACCEPTING THIS COURSEWARE YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. IF YOU DO NOT AGREE YOU MAY RETURN IT TO THE SANS INSTITUTE FOR A FULL REFUND, IF APPLICABLE. The SANS Institute hereby grants User a non-exclusive license to use the material contained in this courseware subject to the terms of this agreement. User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of this publication in any medium whether printed, electronic or otherwise, for any purpose without the express written consent of the SANS Institute. Additionally, user may not sell, rent, lease, trade, or otherwise transfer the courseware in any way, shape, or form without the express written consent of the SANS Institute.

The SANS Institute reserves the right to terminate the above lease at any time. Upon termination of the lease, user is obligated to return all materials covered by the lease within a reasonable amount of time.

SANS acknowledges that any and all software and/or tools presented in this courseware are the sole property of their respective trademark/registered/copyright owners.

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

For518_5_A05_01



Author: Sarah Edwards
oompa@csh.rit.edu
http://twitter.com/iamevltwin
http://twitter.com/sansforensics

This page intentionally left blank.

SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

Website
digital-forensics.sans.org

SIFT Workstation
dfir.to/SANS-SIFT

Join The SANS DFIR Community

- Blog: dfir.to/DFIRBlog
- Twitter: @sansforensics
- Facebook: sansforensics
- Google+: gplus.to/sansforensics
- Mailing list: dfir.to/MAIL-LIST
- YouTube: dfir.to/DFIRCast

DFIR CURRICULUM

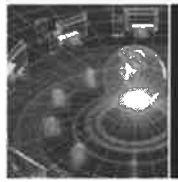
CORE

FOR408
Windows Forensics
GCFE

SEC504
Hacker Techniques, Exploits, and Incident Handling
GCIH

IN-DEPTH INCIDENT RESPONSE

FOR508
Advanced Incident Response
GCFA



FOR572
Advanced Network Forensics and Analysis
GNFA

LEARN REM

FOR610
REM: Malware Analysis
GREM

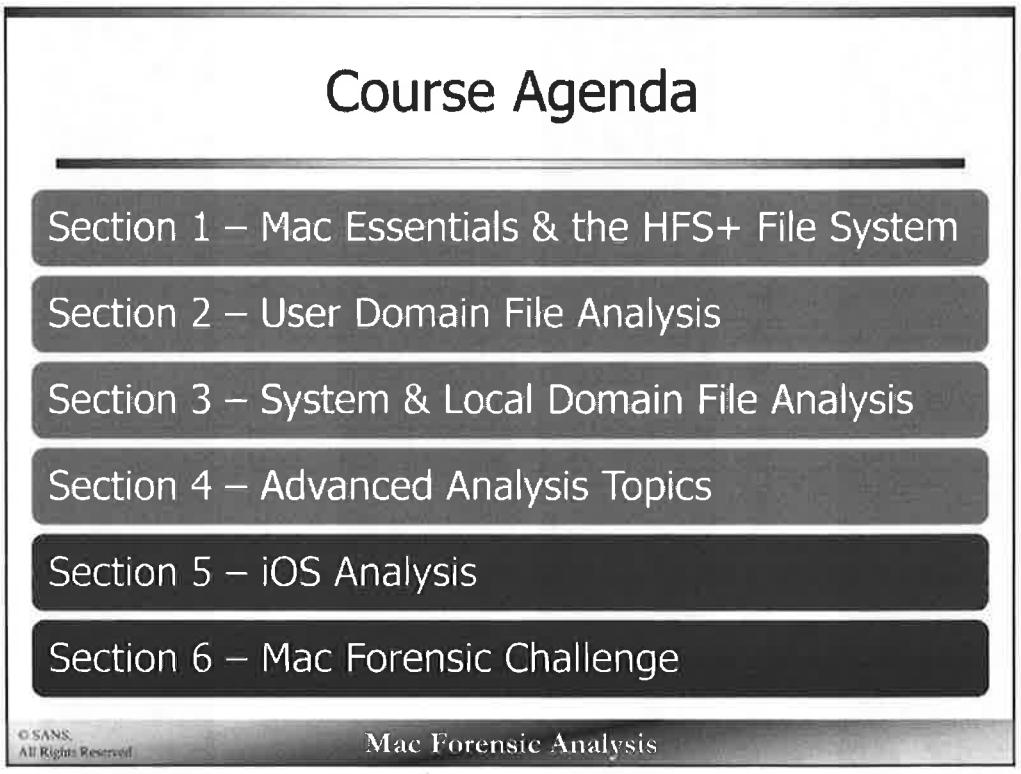
SPECIALIZATION

FOR518
Mac Forensics

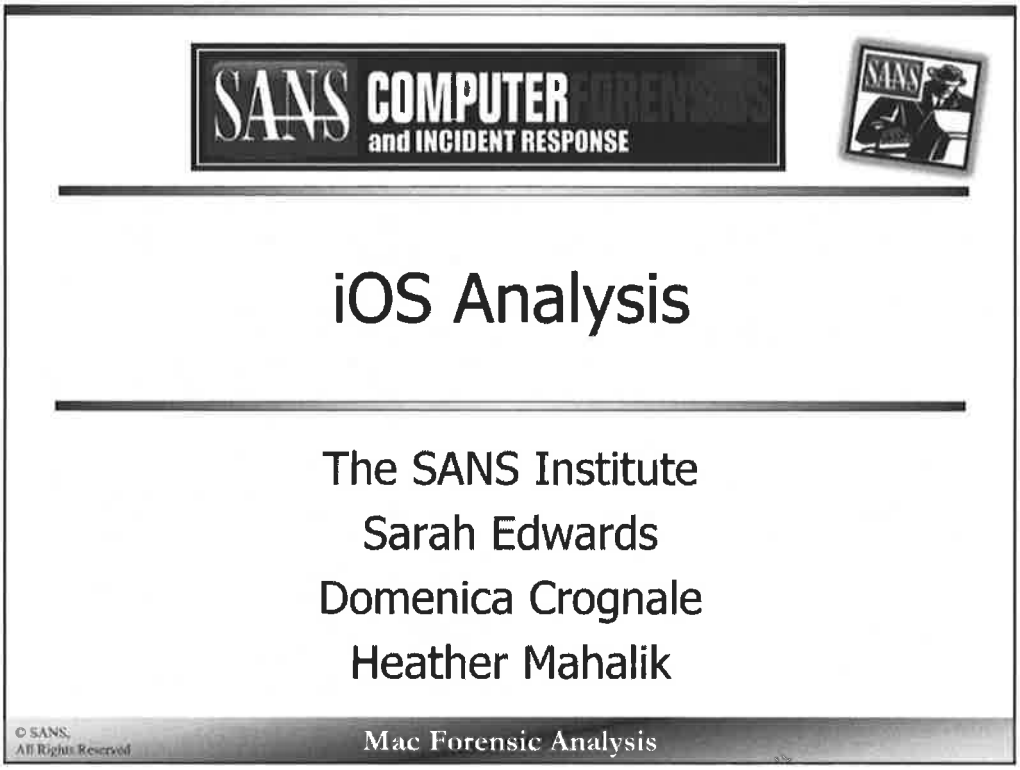
FOR526
Memory Forensics In-Depth

MGT535
Incident Response Team Management

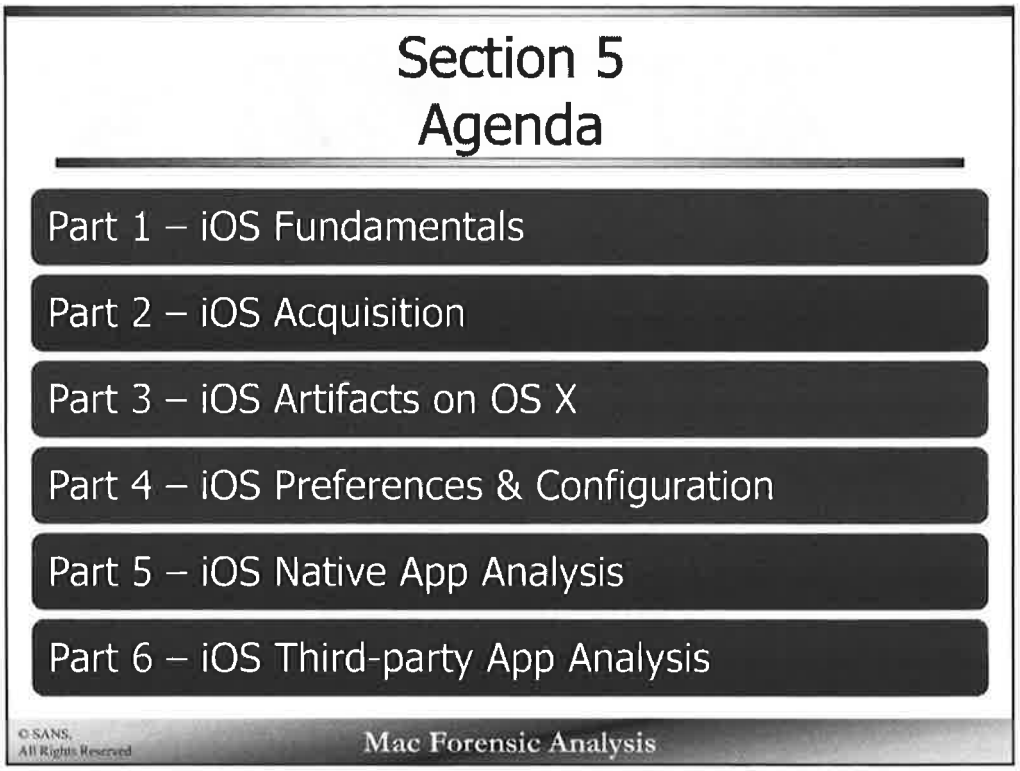
FOR585
Advanced Smartphone Forensics



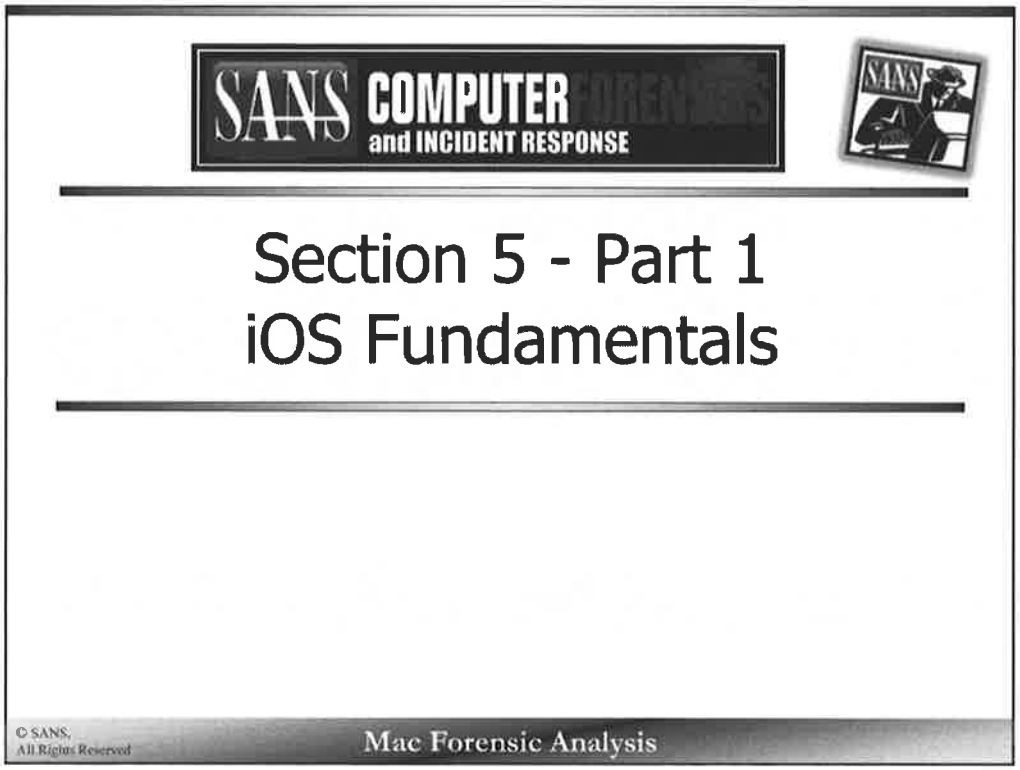
This page intentionally left blank.



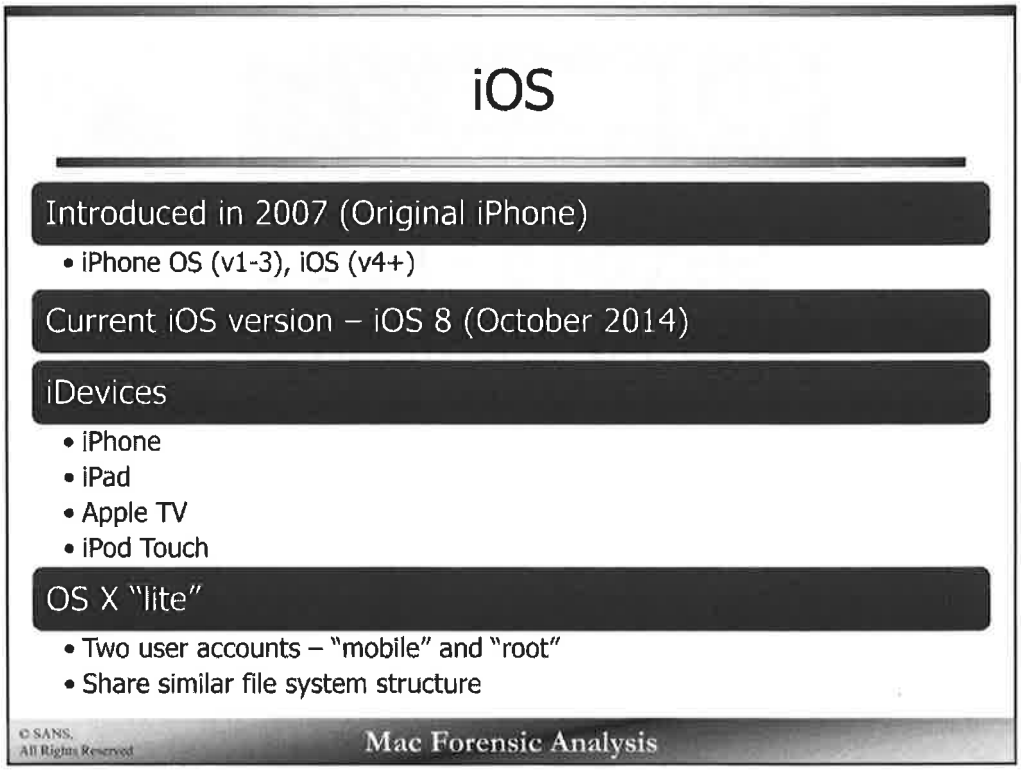
This page intentionally left blank.



This page intentionally left blank.



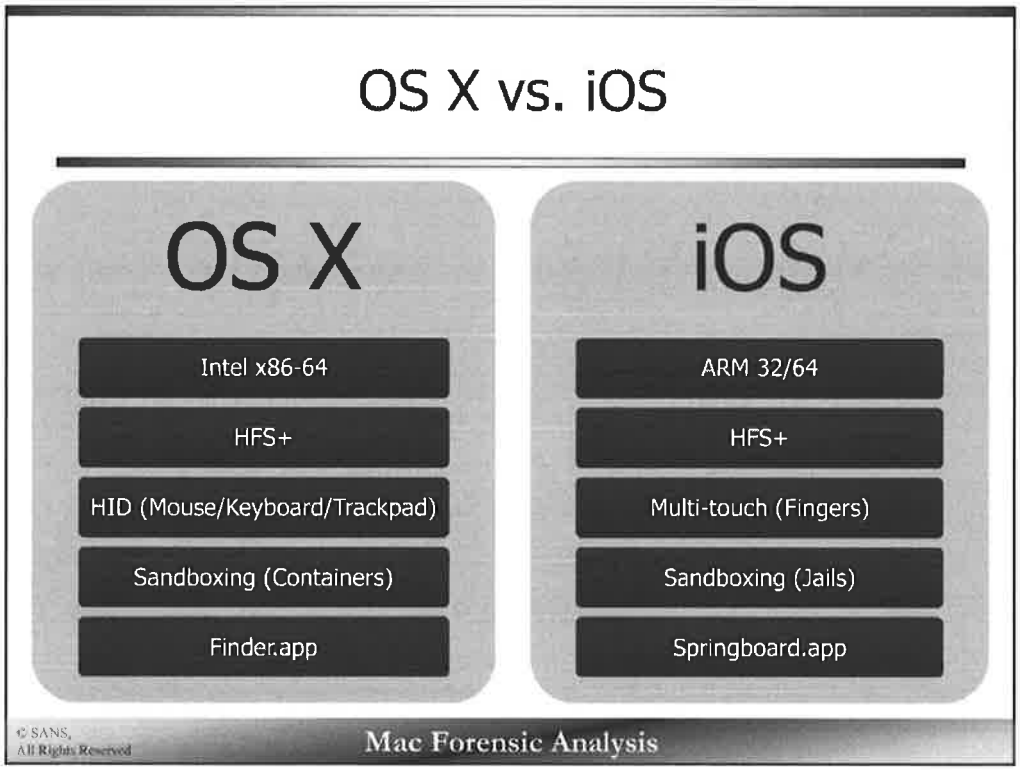
This page intentionally left blank.



iOS was introduced in 2007 with the original iPhone. The original OS was called iPhone OS for the first three operating systems. With the release of the iPod Touch, Apple TV, and iPad the operating system was renamed to iOS in version 4. The current version is iOS 8 released in October 2014.

Most of the mobile devices manufactured by Apple implement iOS such as the iPhone, iPad, iPod Touch, as well as the newer generations of Apple TV. (Fun fact, the original Apple TV implemented the full version of OS X.) Other Apple mobile devices such as the iPod Shuffle, Nano, or Classic use a proprietary embedded operating system which will not be covered in this course.

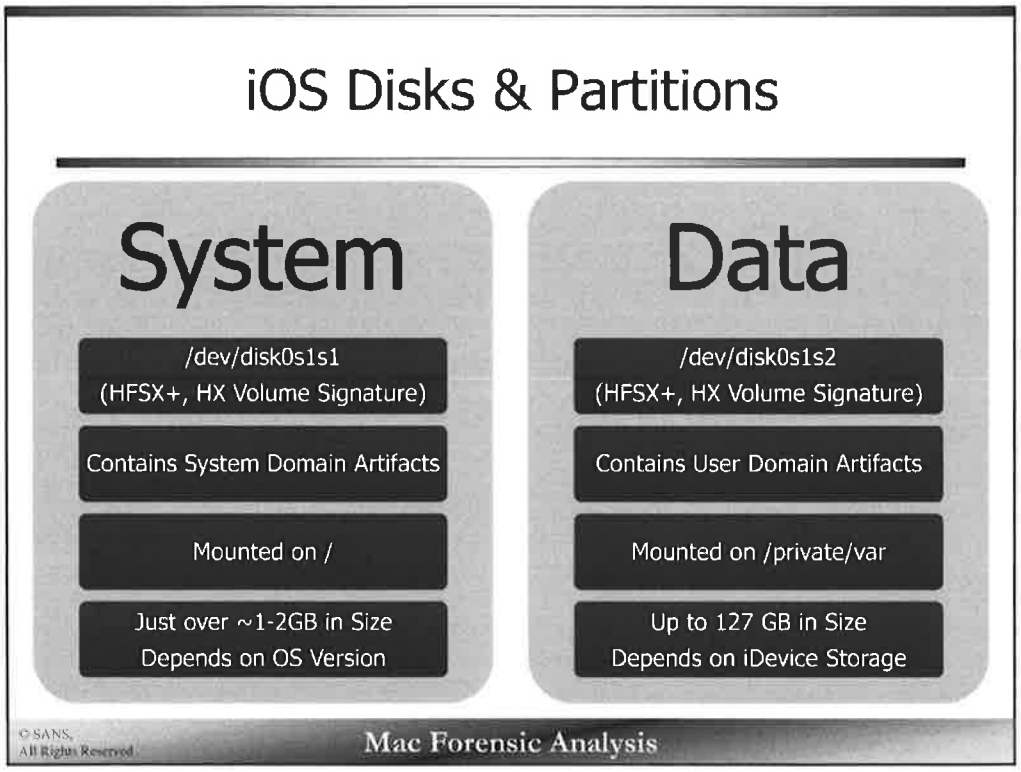
iOS can be thought of as a "lite" version of OS X. Only two user accounts exist on these devices, the standard user "mobile" and the privileged user "root". The default passwords for the "root" and "mobile" accounts is "alpine" if accessed on a jailbroken device.



In newer versions OS X and iOS seem to be merging. Many of the files are similar, they both use sandboxing, implement HFS+, and the file directory structure are alike.

They still different in that they run of different architectures and use different ways present the data to the user and to allow the user to interact with the systems.

HID – Human Interface Device



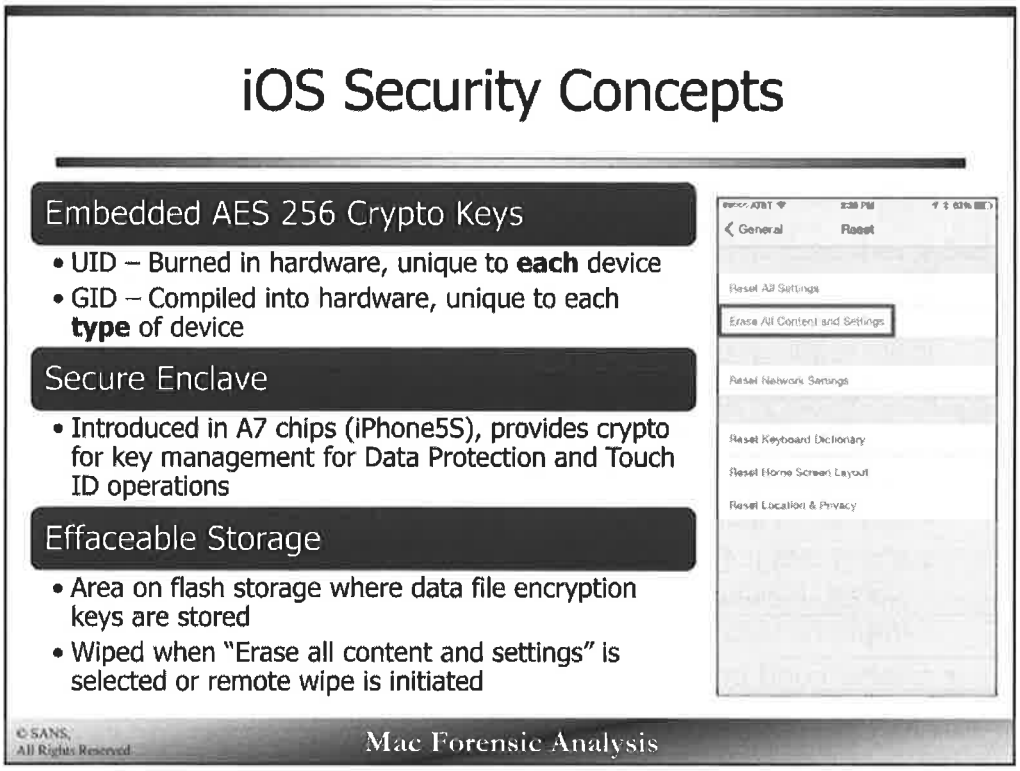
Each iDevice contains two partitions. The System partition contains the system related files and binaries. It is mounted on "/" or the root directory. Depending on the iOS version, it is usually about 1-2 gigabytes in size.

The Data partition is where the user data is stored, all the applications, phone records, photos, etc. It is mounted on /private/var and can be up to 127 gigabytes in size, depending on the size of the device.

There are no external storage areas on the device other an the SIM card.

Reference:

http://theiphonewiki.com/wiki//private/etc/fstab



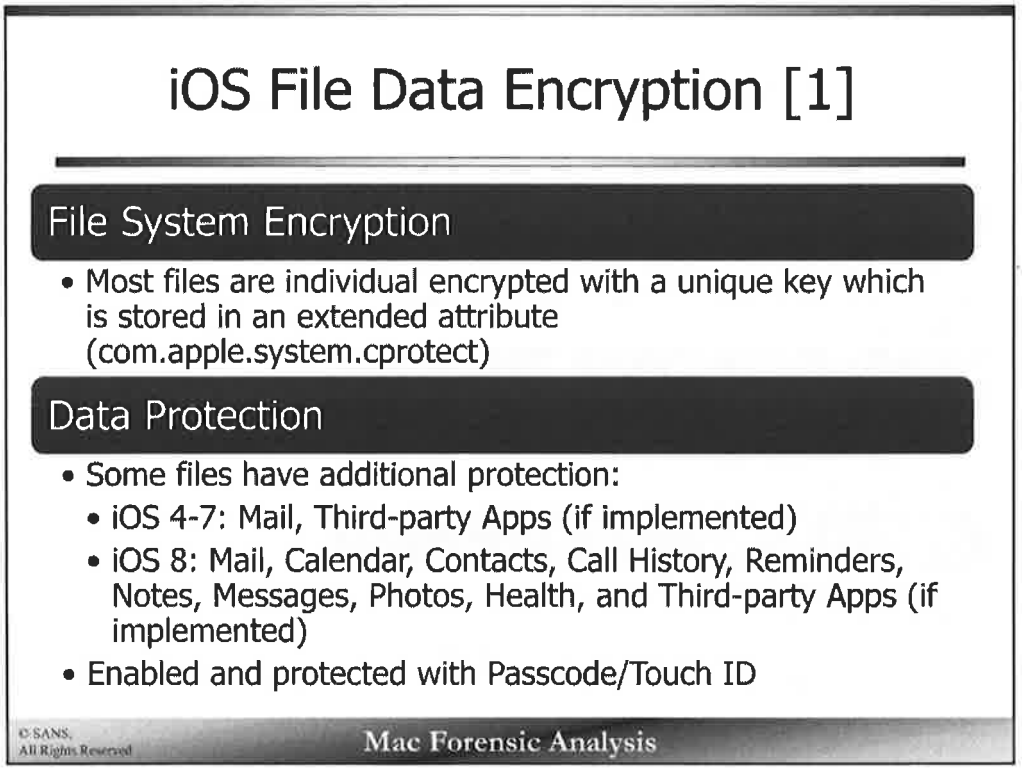
Cryptographic keys are burned or compiled into the physical hardware of the device. The Secure Enclave coprocessor provides the cryptographic processing using these keys for file system encryption, Data Protection as well as usage of Touch ID. The Secure Enclave was introduced in the A7 chips in the iPhone 5S generation of devices.

These keys are used to cryptographically protect data on the device by using them as keys to create other keys to encrypt specific data files. These additional keys are stored in the Effaceable Storage area.

This storage area can be wiped by the user when they select the "`Erase all content and settings`" option available in the `Settings | General | Reset` menu. This activity resets the device to where a forensic analyst will not be able to recover any user data because it is encrypted by the keys that have just been wiped. The same functionality is performed with a user performs a remote wipe using MDM software or iCloud.

References:

Apple iOS Security White Paper: https://www.apple.com/privacy/docs/iOS_Security_Guide_Oct_2014.pdf



A large portion of the files located on the Data partition of an iOS device implement the "`NSFileProtectionNone`" class key – meaning they are only encrypted with the burned-into-hardware UID key. This class key is stored in the Effaceable Storage area while the file key is stored in the `com.apple.system.cprotect` extended attribute for the file.

Data Protection is used to give extra protection for various file classes. Starting in iOS4, the Mail application received this additional security. This additional security is enabled by use of a passcode and/or Touch ID. Some of the class keys that implement this protection are `NSFileProtectionComplete`, `NSFileProtectionCompleteUnlessOpen`, and `NSFileProtectionCompleteUntilFirstUserAuthentication`).These class keys require the users passcode and/or TouchID for access. This is why a some data may not be accessible in a forensic acquisition without the users credentials.

Encryption of file data

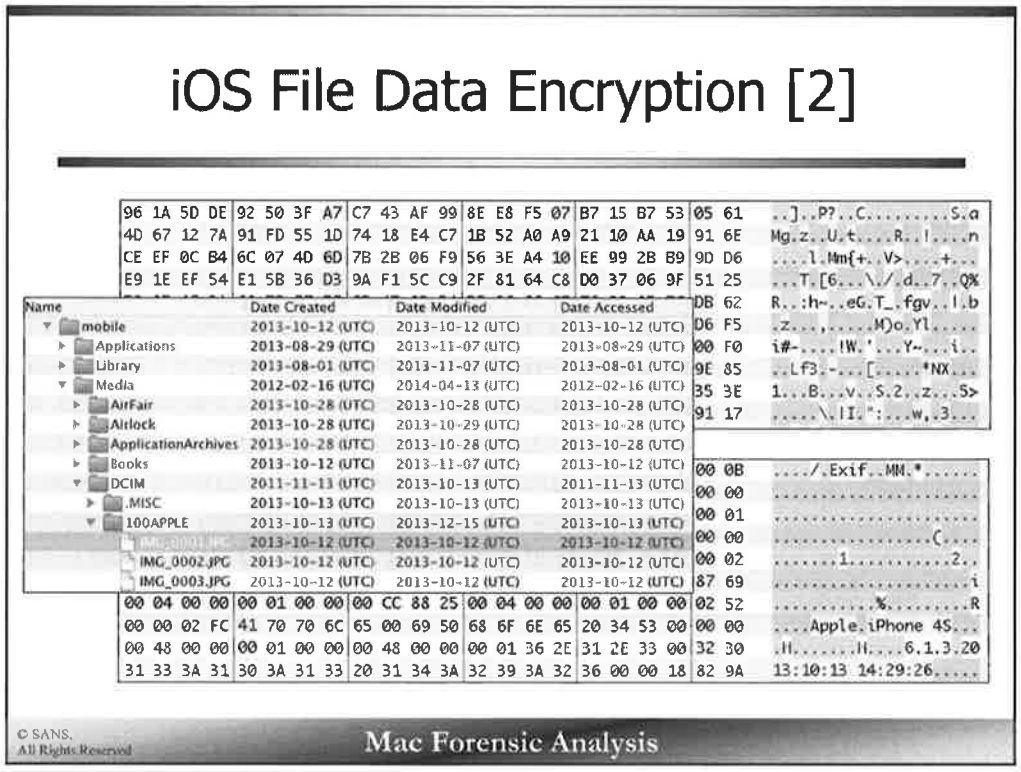
iOS 4-7: Mail, Third-party Apps (if implemented)

iOS 8: Mail, Calendar, Contacts, Call History, Reminders, Notes, Messages, Photos, Health, and Third-party Apps (if implemented)

References:

Apple iOS Security White Paper: https://www.apple.com/privacy/docs/iOS_Security_Guide_Oct_2014.pdf

iPhone Data Protection In Depth: http://esec-lab.sogeti.com/dotclear/public/publications/11-hitbamsterdam-iphonedataprotection.pdf



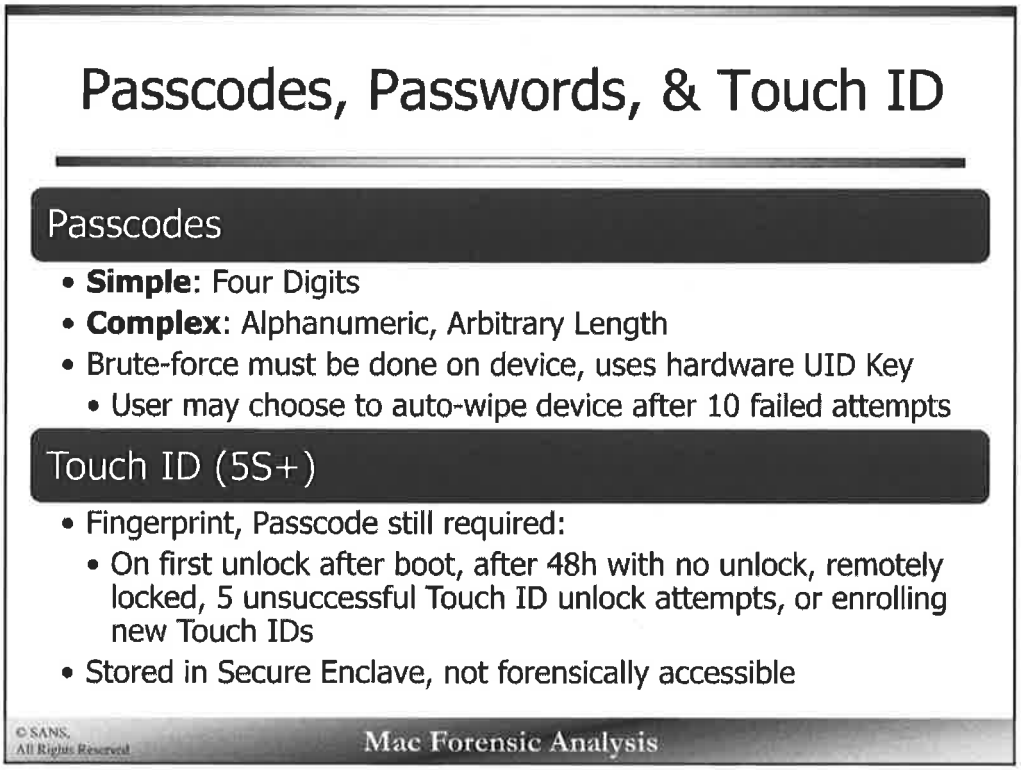
The hardware encryption is shown above, the top hex editor screenshot shows the encrypted contents of the `IMG_001.JPG` file highlighted in the Blacklight screenshot. After file decryption, we can see the unencrypted contents of this file.

Because the file encryption is performed on a per-file basis, the file system structure and metadata is still accessible, however the contents of the files are not.

Name	Date Created	Date Modified	Date Accessed
mobile	2013-10-12 (UTC)	2013-10-12 (UTC)	2013-10-12 (UTC)
Applications	2013-08-29 (UTC)	2013-11-07 (UTC)	2013-08-29 (UTC)
Library	2013-08-01 (UTC)	2013-11-07 (UTC)	2013-08-01 (UTC)
Media	2012-02-16 (UTC)	2014-04-13 (UTC)	2012-02-16 (UTC)
AirFair	2013-10-28 (UTC)	2013-10-28 (UTC)	2013-10-28 (UTC)
Airlock	2013-10-28 (UTC)	2013-10-29 (UTC)	2013-10-28 (UTC)
ApplicationArchives	2013-10-28 (UTC)	2013-10-28 (UTC)	2013-10-28 (UTC)
Books	2013-10-12 (UTC)	2013-11-07 (UTC)	2013-10-12 (UTC)
DCIM	2011-11-13 (UTC)	2013-10-13 (UTC)	2011-11-13 (UTC)
.MISC	2013-10-13 (UTC)	2013-10-13 (UTC)	2013-10-13 (UTC)
100APPLE	2013-10-13 (UTC)	2013-12-15 (UTC)	2013-10-13 (UTC)
IMG_0001.JPG	2013-10-12 (UTC)	2013-10-12 (UTC)	2013-10-12 (UTC)
IMG_0002.JPG	2013-10-12 (UTC)	2013-10-12 (UTC)	2013-10-12 (UTC)
IMG_0003.JPG	2013-10-12 (UTC)	2013-10-12 (UTC)	2013-10-12 (UTC)

```
96 1A 5D DE 92 50 3F A7 C7 43 AF 99 8E E8 F5 07 B7 15 B7 53 05 61  ..]..P?..C.........S.a
4D 67 12 7A 91 FD 55 1D 74 18 E4 C7 1B 52 A0 A9 21 10 AA 19 91 6E  Mg.z..U.t....R..!....n
CE EF 0C B4 6C 07 4D 6D 7B 2B 06 F9 56 3E A4 10 EE 99 2B B9 9D D6  ....l.Mm{+..V>....+...
E9 1E EF 54 E1 5B 36 D3 9A F1 5C C9 2F 81 64 C8 D0 37 06 9F 51 25  ...T.[6...\./.d..7..Q%
52 8B 1E 3A 68 7E E7 BF 65 47 8D 54 5F 0C 66 67 76 09 85 7C DB 62  R..:h~..eG.T_.fgv..|.b
05 7A 81 E3 A6 2C CE 8A F2 FF C9 4D 29 6F FD 59 6C 1D 17 98 D6 F5  .z...,.....M)o.Yl.....
69 23 2D 86 E0 9E 0C 21 57 B0 27 AD 1D C5 59 7E CC DA 85 69 00 F0  i#-....!W.'...Y~...i..
B8 C0 4C 66 33 D2 2D 0E D6 C6 5B F0 93 EF 0B E9 2A 4E 58 FD 9E 85  ..Lf3.-...[.....*NX...
31 BB EE 1C 42 86 14 FB 76 EE 90 53 F6 32 E3 1A 7A 9D 99 85 35 3E  1...B...v..S.2..z...5>
EE EC FE 0E C7 5C 11 7C 49 C7 22 3A B0 EC A6 77 2C D8 33 9F 91 17  .....\.|I.":...w,.3...
```

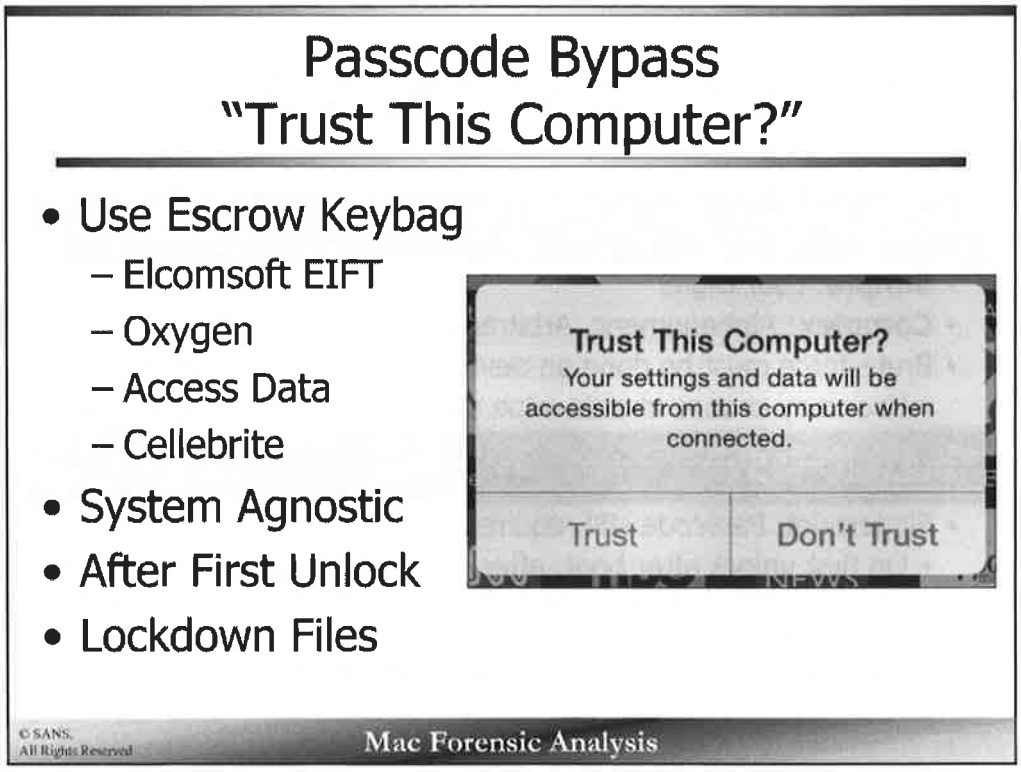
```
FF D8 FF E1 2F FE 45 78 69 66 00 00 4D 4D 00 2A 00 00 00 08 00 0B  ..../.Exif..MM.*......
01 0F 00 02 00 00 00 06 00 00 00 92 01 10 00 02 00 00 00 0A 00 00  ......................
00 98 01 12 00 03 00 00 00 01 00 06 00 00 01 1A 00 05 00 00 00 01  ......................
00 00 00 A2 01 1B 00 05 00 00 00 01 00 00 00 AA 01 28 00 03 00 00  ...................(....
00 01 00 02 00 00 01 31 00 02 00 00 00 06 00 00 00 B2 01 32 00 02  .......1...........2..
00 00 00 14 00 00 00 B8 02 13 00 03 00 00 00 01 00 01 00 00 87 69  ......................i
00 04 00 00 00 01 00 00 00 CC 88 25 00 04 00 00 00 01 00 00 02 52  ...........%.........R
00 00 02 FC 41 70 70 6C 65 00 69 50 68 6F 6E 65 20 34 53 00 00 00  ....Apple.iPhone 4S...
00 48 00 00 00 01 00 00 00 48 00 00 00 01 36 2E 31 2E 33 00 32 30  .H.......H....6.1.3.20
31 33 3A 31 30 3A 31 33 20 31 34 3A 32 39 3A 32 36 00 00 18 82 9A  13:10:13 14:29:26.....
```



There are many different methods to protect an iDevice.

There are two types of passcodes that can be used. A simple passcode is a four digit passcode, or a complex passcode which can be made up of alphanumeric characters of any length.

On newer devices, a user may implement TouchID. Touch ID is used along with a passcode to allow a user to unlock their phone with their fingerprint. The passcode is still required in certain circumstances such as after the phone starts up or after a reboot.



Starting with iOS 7, once a device has been connected to a system a "`Trust This Computer?`" windows appears on the device. When the user selects "`Trust`" the lockdown files are created in the following locations.

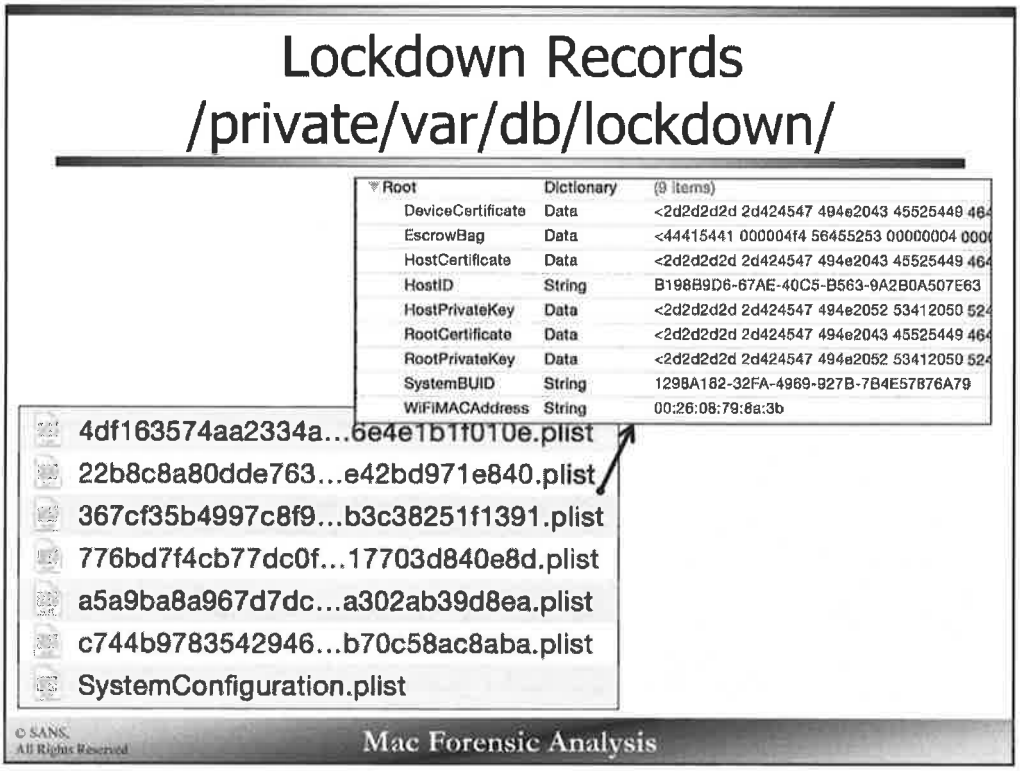
OS X - `/private/var/db/lockdown/`

Windows XP - `\Documents and Settings\<user>\Application Data\Apple Computer\Lockdown\`

Windows Vista - `\Users\<user>\AppData\Roaming\Apple Computer\Lockdown\`

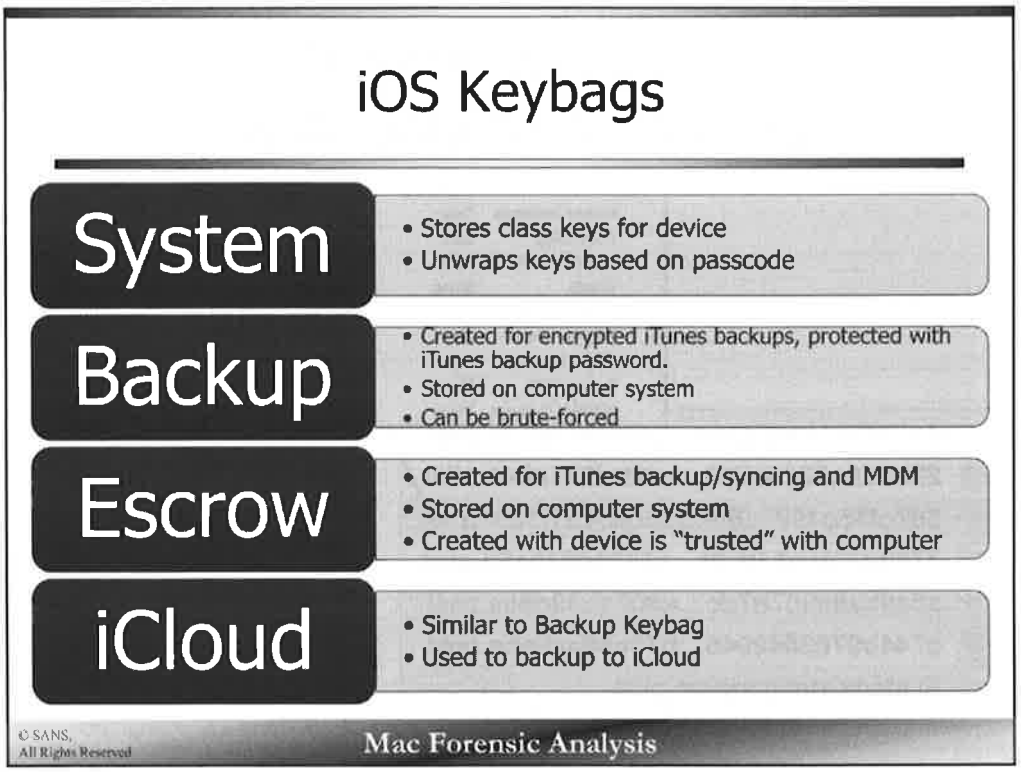
Windows 7+ - `\Program Data\Apple\Lockdown\`

Since these files are system agnostic, they may be copied to your analysis system or input into various acquisition software and used to access the iDevice.

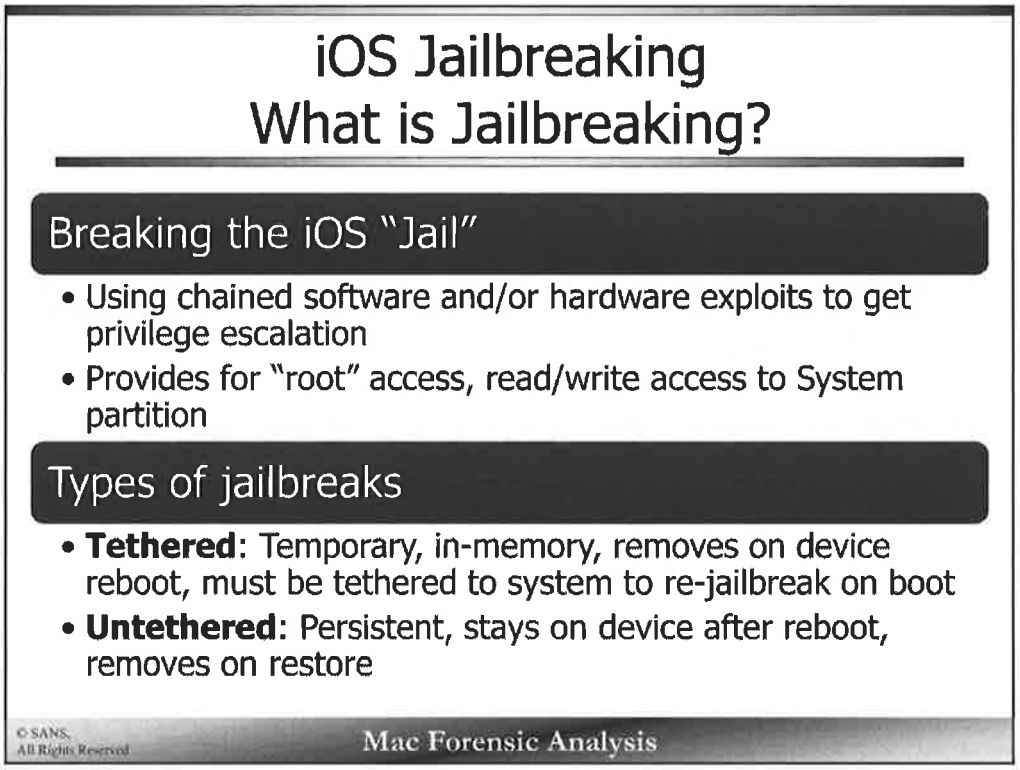


The lockdown files on OS X are plist files. Each plist file is named with the UDID of the device that has been trusted.

In the screenshot above, this system has trusted six different iDevices. Each plist contains various certificate, keybag, and other information that can be used to access a locked device.



There are four types of iOS keybags; System, Backup, Escrow, and iCloud described above.



The jailbreaking process allows a user to escalate their privileges on an iOS device using chained software and/or hardware exploits. Jailbreaking a device allows a user root access to the System partition with read and write access, as oppose to only read-only access.

There are two main types of jailbreaks; tethered and untethered. A tethered jailbreak is a temporary (in-memory) jailbreak that requires the devices to be "tethered" to a system in order to keep the jailbreak through the reboot process. A untether jailbreak is a persistent jailbreak that allows the user to reboot as they with without the need for another system. This jailbreak can be removed by doing a restore through iTunes.

References:

http://theiphonewiki.com/wiki/Jailbreak

http://theiphonewiki.com/wiki/Tethered_jailbreak

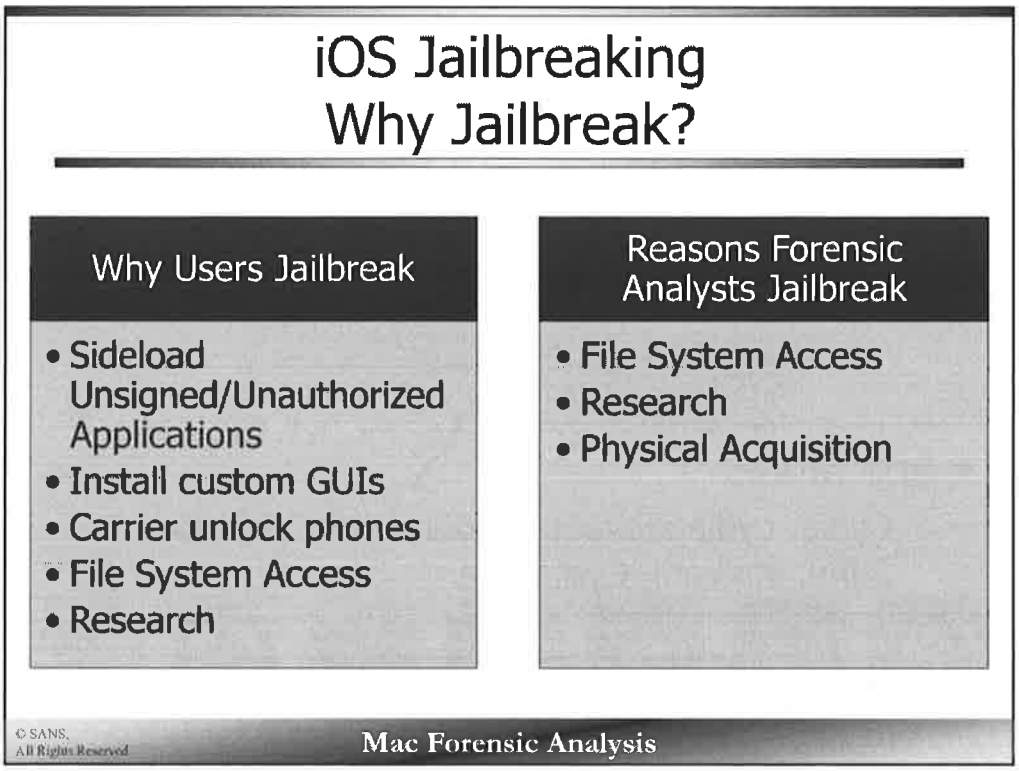
http://theiphonewiki.com/wiki/Untethered_jailbreak

The jailbreak process is very dependent on device hardware and iOS version – down to the point releases. The specific device compatibility changes so frequently with updates from Apple and updates from the jailbreak hackers that it is best to determine if the device that you have in hand can be jailbroken. The charts on http://theiphonewiki.com/wiki/Jailbreak are updated frequently.

Popular software used for newer devices include; pangu, taiG evasi0n, p0sixpwn, redsn0w, and sn0wbreeze. There are many different software packages available, if you refer to the charts listed on theiphonewiki.com.

References:

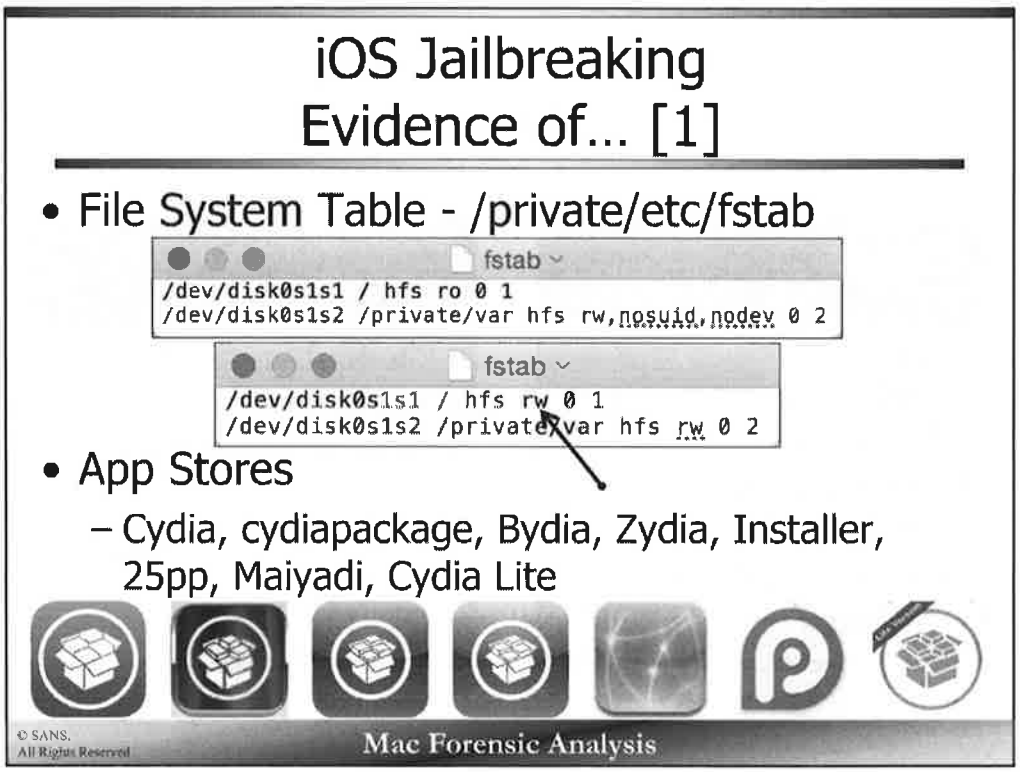
http://theiphonewiki.com/wiki/Jailbreak



Users jailbreak for many reasons:

- To install an application that is not an authorized app in the Apple App Store
- Install custom graphical user interfaces
- Unlock various carrier-based locks – Users may want to unlock their phones to be able to use them on other networks
- Access to the file system – Users may want to access some application data files or upload their own files
- Research – Hackers, developers, and/or researchers may use jailbreak for research purposes

As forensic analysts we may also choose to jailbreak a device, some for the same reasons as users. We may need file system access for research purposes (i.e.: what data does this app store) and to acquire a users device in a physical format. Newer iOS devices require a forensic analyst to jailbreak the device to access the System partition, or to acquire deleted/unallocated data from the User partition.

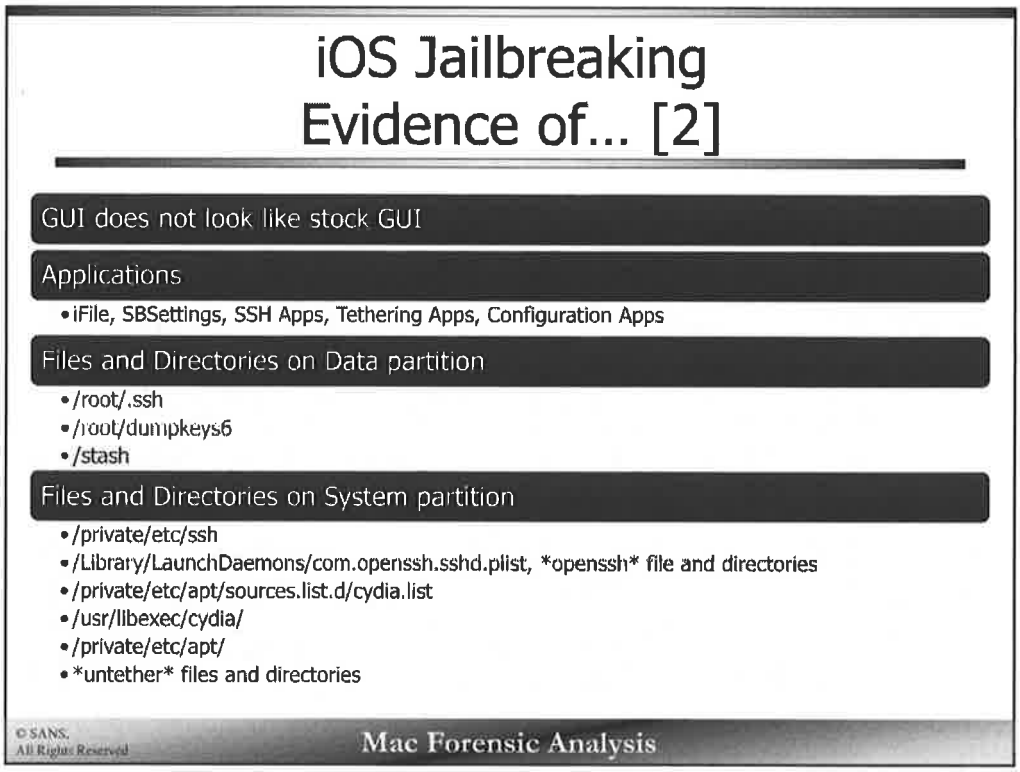


The file system table located at `/private/etc/fstab` can be used to determine if a device has been jailbroken. This file is located on the System partition. This file shows how each partition is mounted.

- `/dev/disk0s1s1` – System Partition
- `/dev/disk0s1s2` – Data Partition (User)

The two screenshots above show two different devices; the top screenshot shows a non-jailbroken device, while the bottom screenshot shows a jailbroken device. Notice the "`rw`" and the "`ro`" mounting options. The bottom example shows the `System` partition was mounted as `rw` - read/write (Jailbroken!), while the bottom shows the `System` partition was mounted as `ro` - read-only (Default). It should be noted here that not all jailbreaking software will mount the System partition as read/write.

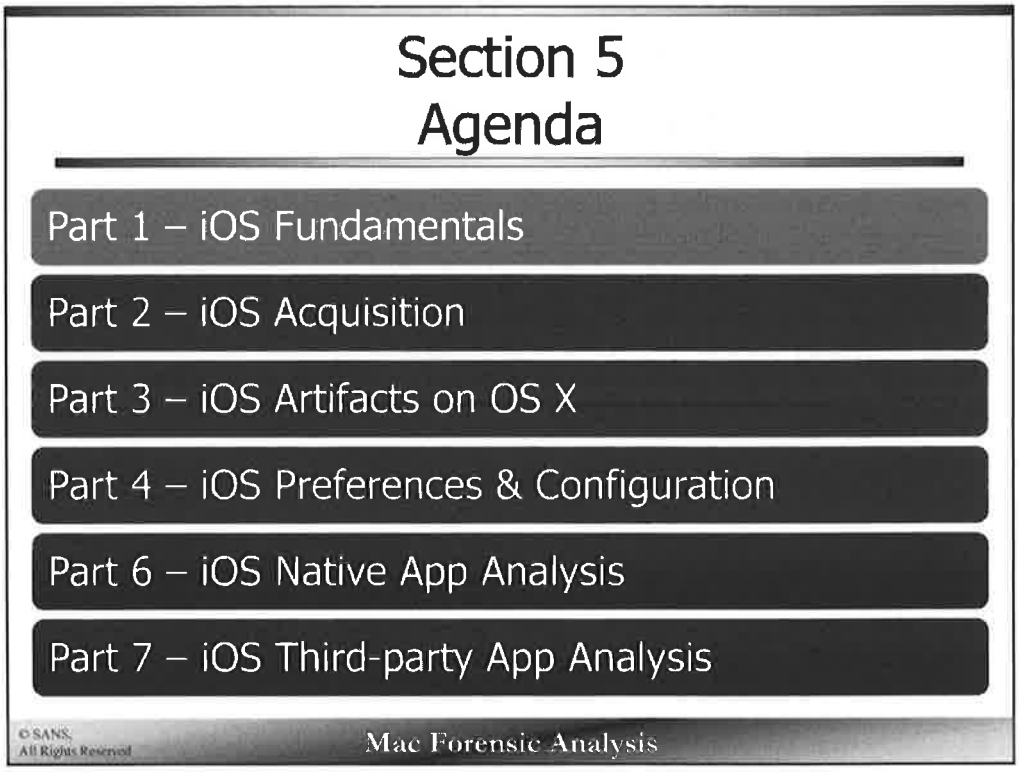
The presence of unofficial app stores can also show that a device has been jailbroken. The main unofficial app store used in the US is Cydia which downloads its applications into `/Applications` on the device. Be on the lookout for other lesser known app stores or icons that play off of the Cydia app icon.



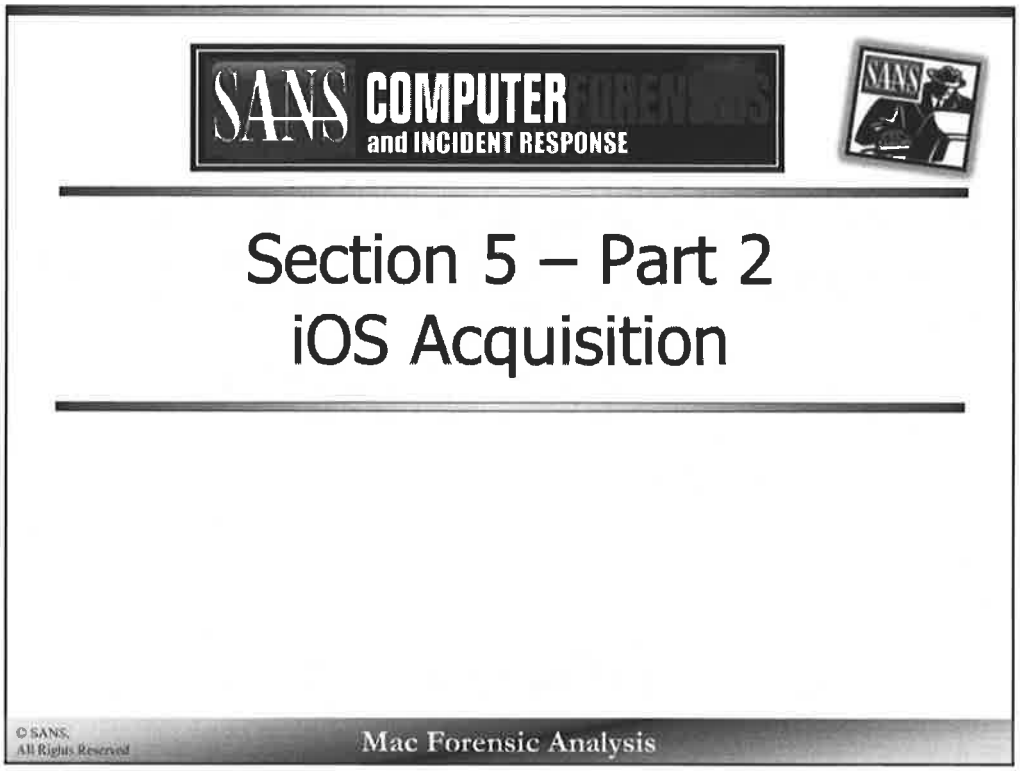
You may also tell if the device has been jailbroken by the way it looks, look for non-stock icons, backgrounds, dock bar, notification center, etc.

Certain popular non authorized applications may be installed like `iFile` and `SBSettings`, but all be on the lookout for SSH, tethering, or configuration applications.

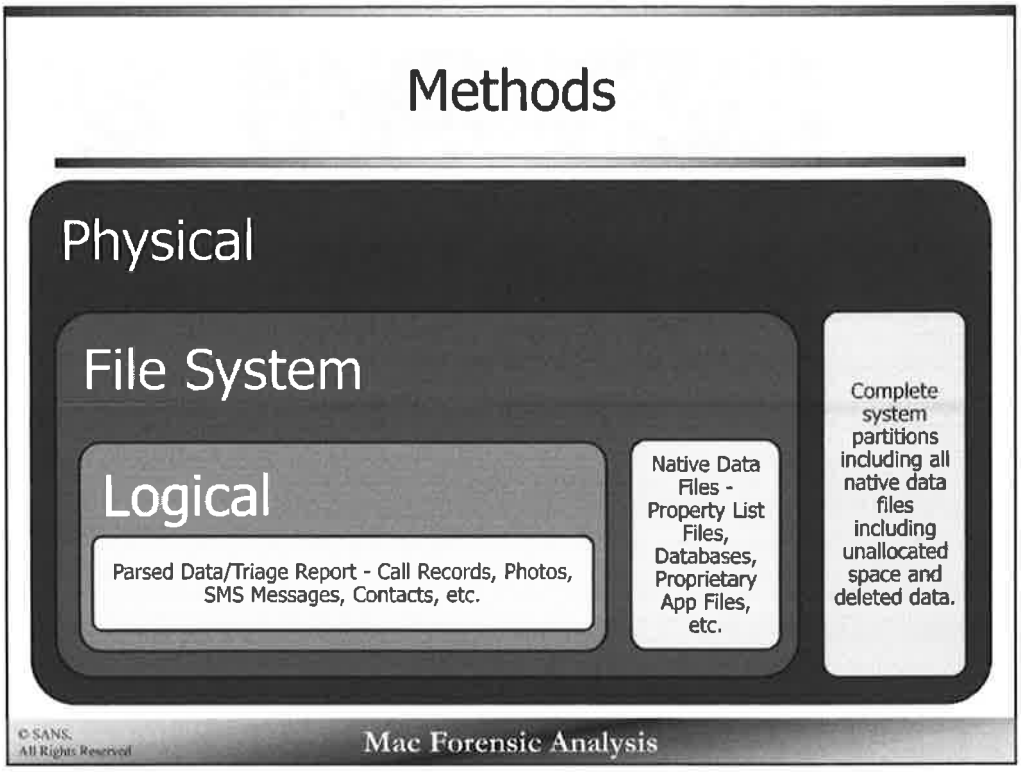
Once an image is acquired, look at the file system on each partition – it should be obvious that is was jailbroken by looking at various directories.



This page intentionally left blank.



This page intentionally left blank.

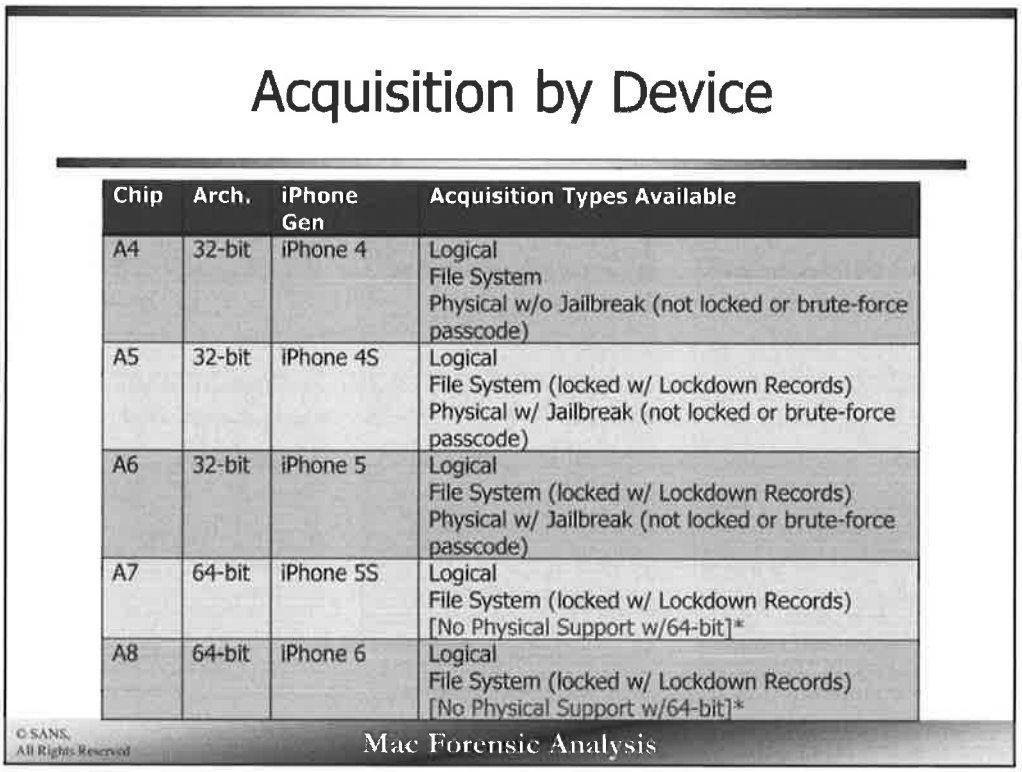


There are three different types of mobile acquisitions.

The most basic, is the Logical acquisition. This type only includes basic data that is parsed by the tools such as phone records, extracted photos, SMS messages, contact information. This will not include deleted data or whole database files that you can parse.

A File System Acquisition collects logical database, plist, photos, and other related application data. You will get lots of user data in their original database or files but you will not be deleted (in free space) data. You will also not get all files on the system – some files cannot be acquired due to permissions on the Data partition or the whole System partition.

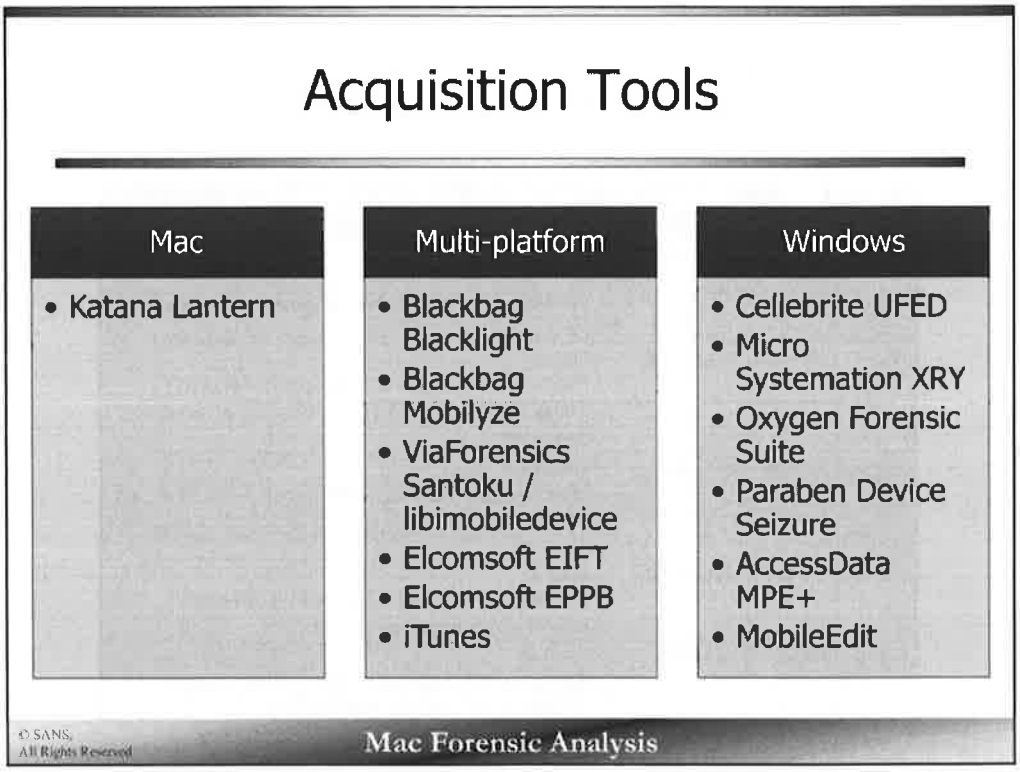
A complete Physical Acquisition will provide you with all the files, deleted data, and system level data. This is a bit-by-bit copy of the file system.



The table above gives a general overview of what can be acquired using using what means.

Please note each acquisition tool is different, please refer to their documentation for specifics.

* These devices can be jailbroken and their contents copied via SSH, SFTP, AFC2 software, or other non-forensic means. You may be able to copy the partitions, but be aware that some files will have Data Protection implemented and you may not be able to decrypt the contents of the file.

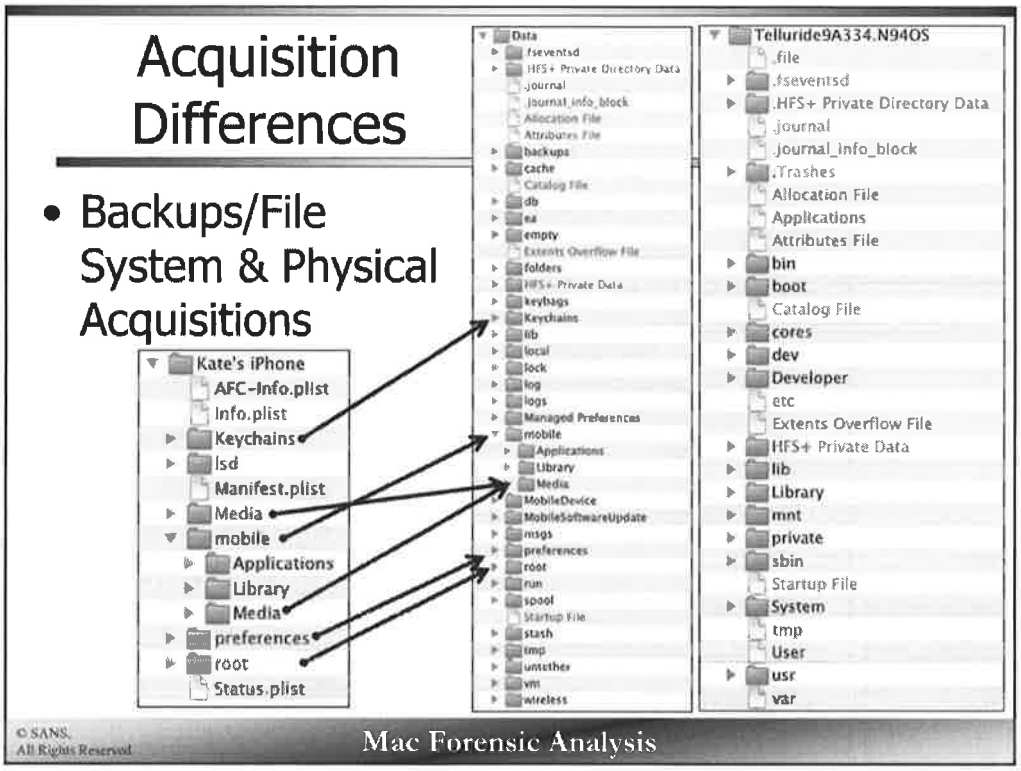


There are many tools available that allow an investigator to acquire an iDevice.

The tools differ in what they can acquire, how they can acquire, and what platform they run on.

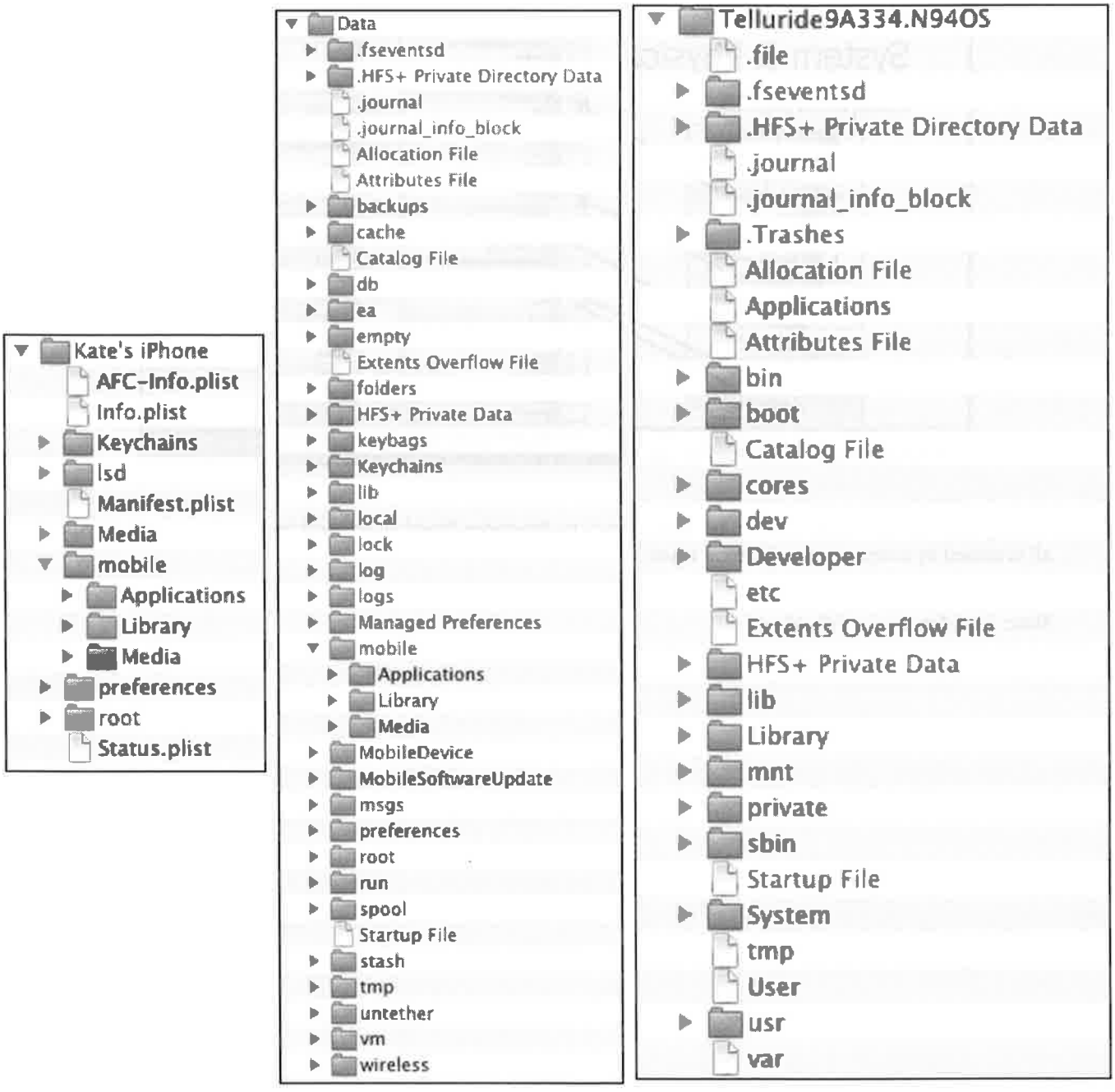
If you have an older device, these mostly deprecated tools may be an option. (Acquisition of iPhone 4 generation & older):

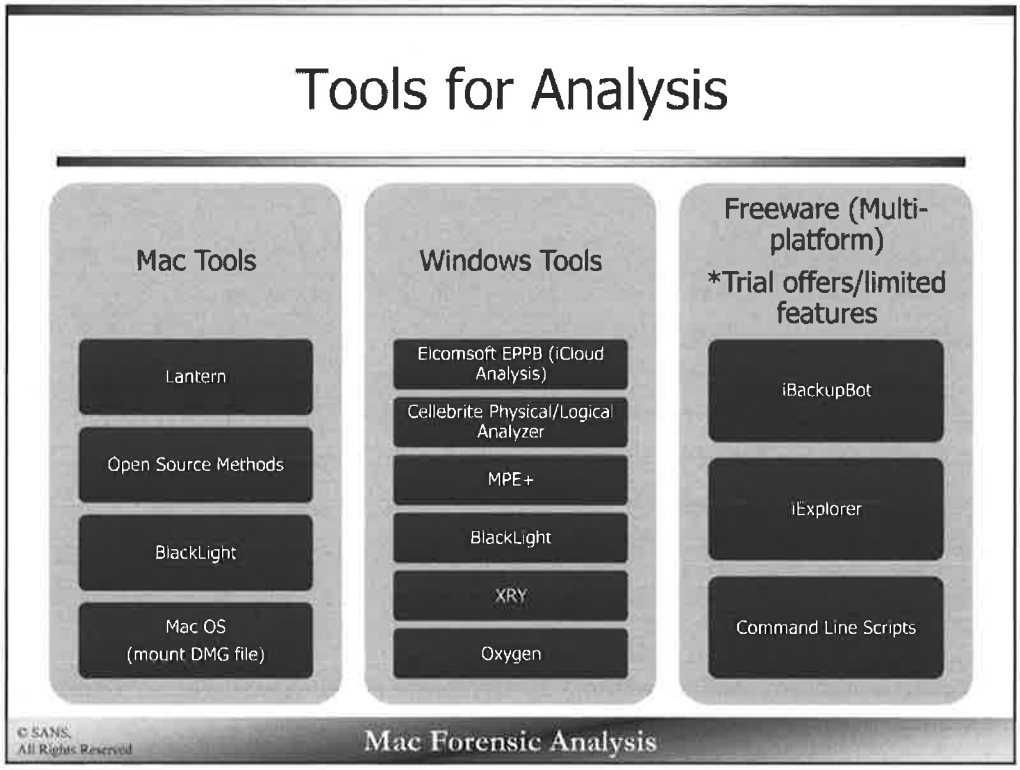
- iXAM - http://ixam-forensics.com
- iPhone Analyzer- http://sourceforge.net/projects/iphoneanalyzer/
- iphone-dataprotection - https://code.google.com/p/iphone-dataprotection/
- Zdziarski Method - http://www.iosresearch.org



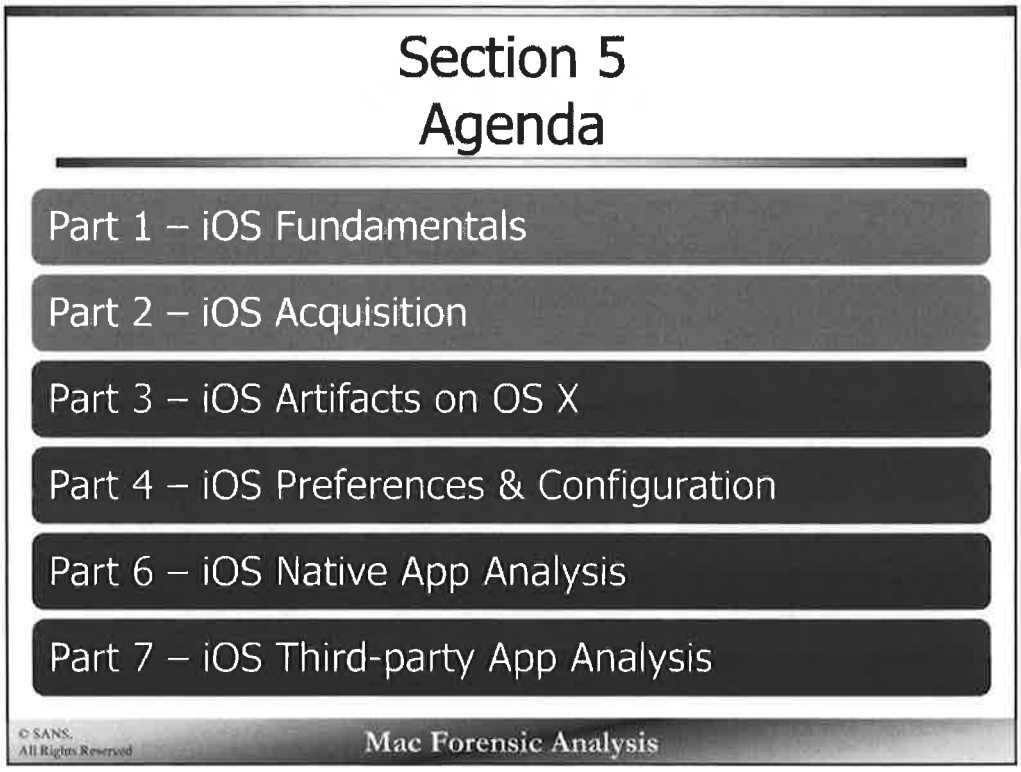
Backups/File system extractions can be mapped back to directories found on physical images. You can see what all is missed by comparing the different types.

Note: `lsd` directory = `/db/lsd/`

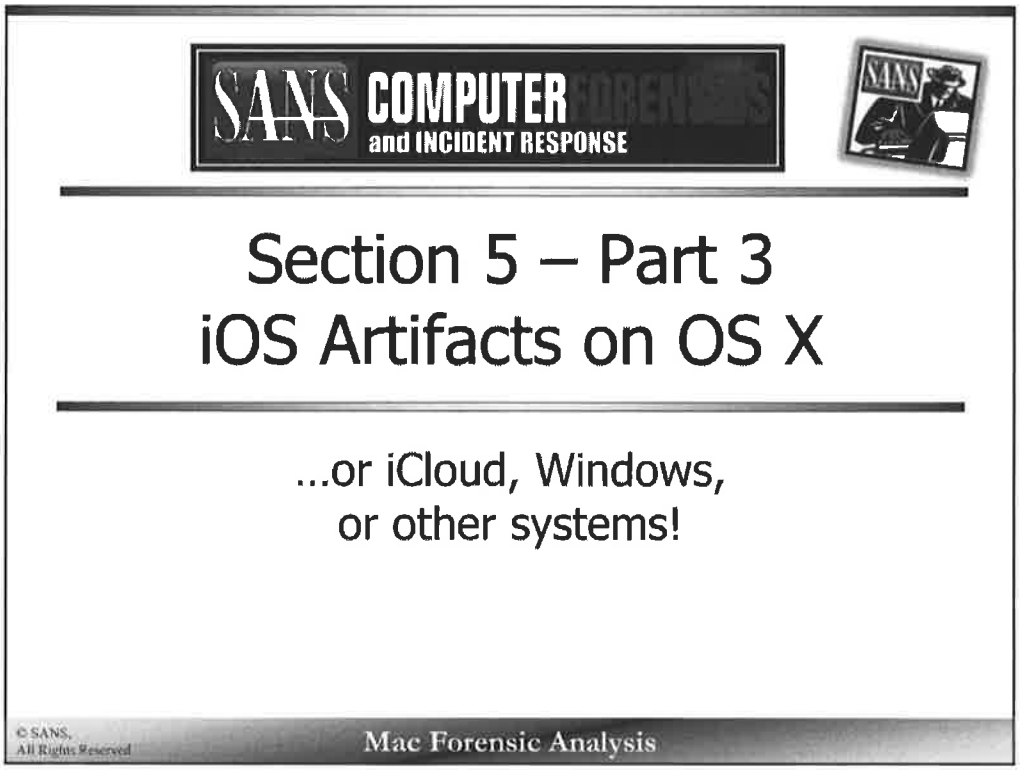

Kate's iPhone
AFC-Info.plist
Info.plist
Keychains
lsd
Manifest.plist
Media
mobile
Applications
Library
Media
preferences
root
Status.plist
Data
.fseventsd
.HFS+ Private Directory Data
.journal
.journal_info_block
Allocation File
Attributes File
backups
cache
Catalog File
db
ea
empty
Extents Overflow File
folders
HFS+ Private Data
keybags
Keychains
lib
local
lock
log
logs
Managed Preferences
mobile
Applications
Library
Media
MobileDevice
MobileSoftwareUpdate
msgs
preferences
root
run
spool
Startup File
stash
tmp
untether
vm
wireless
Telluride9A334.N94OS
.file
.fseventsd
.HFS+ Private Directory Data
.journal
.journal_info_block
.Trashes
Allocation File
Applications
Attributes File
bin
boot
Catalog File
cores
dev
Developer
etc
Extents Overflow File
HFS+ Private Data
lib
Library
mnt
private
sbin
Startup File
System
tmp
User
usr
var



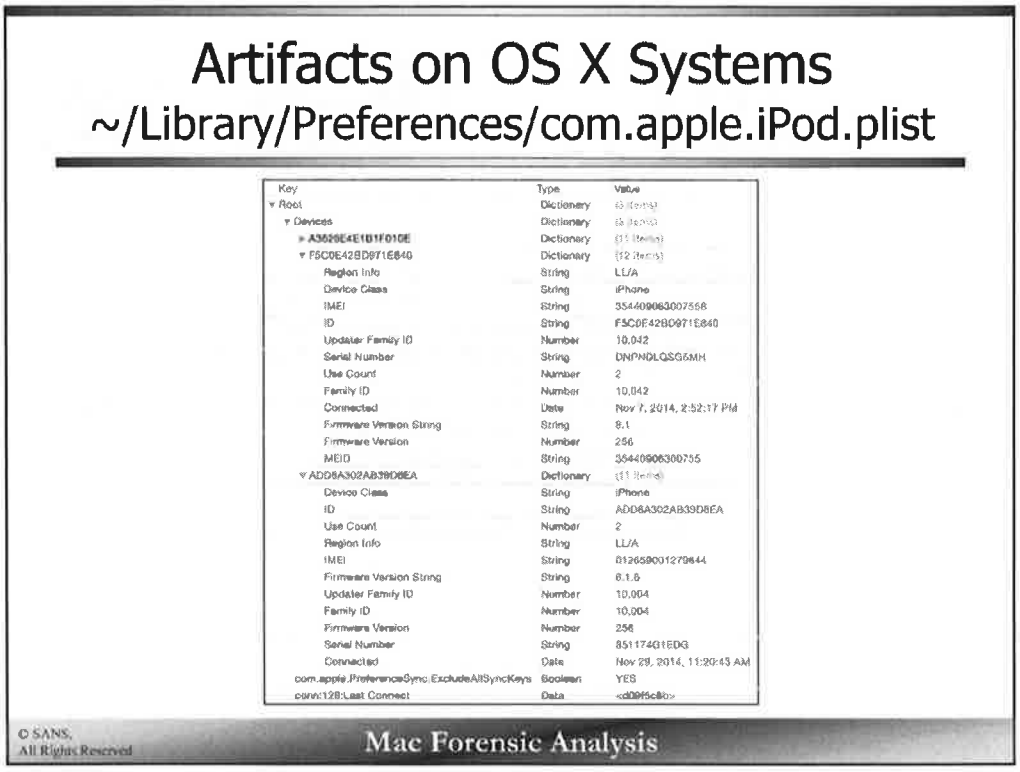
Just as there are multiple tools for iOS acquisition, there are multiple tools that run on different platforms available for iOS analysis. The biggest difference between the tools is the physical image file that is created as a result of a full physical acquisition of a device. Mac-based tools will create a raw image in the form of a disk image or .dmg file, and Windows-based tools will create a raw binary file or .bin.. Analytical tools built for the Mac OS will require that the raw image file is in the form of a .dmg.



This page intentionally left blank.



This page intentionally left blank.



The `com.apple.iPod.plist` file located in the user's preferences directory contains all the iDevices attached to the system while logged in as that user.

While the file is called `com.apple.iPod.plist`, it will contain information for iPods, iPads, and iPhones. Each 16-character alphanumeric key under Devices will contain the information for one device to include:

- Device Class – The type of iDevice connected
- IMEI/MEID – Unique equipment identifiers
- Use Count – Number of times this device was connected
- Connected – The last time this device was connected
- Firmware Version String – The iOS version the device was when last connected

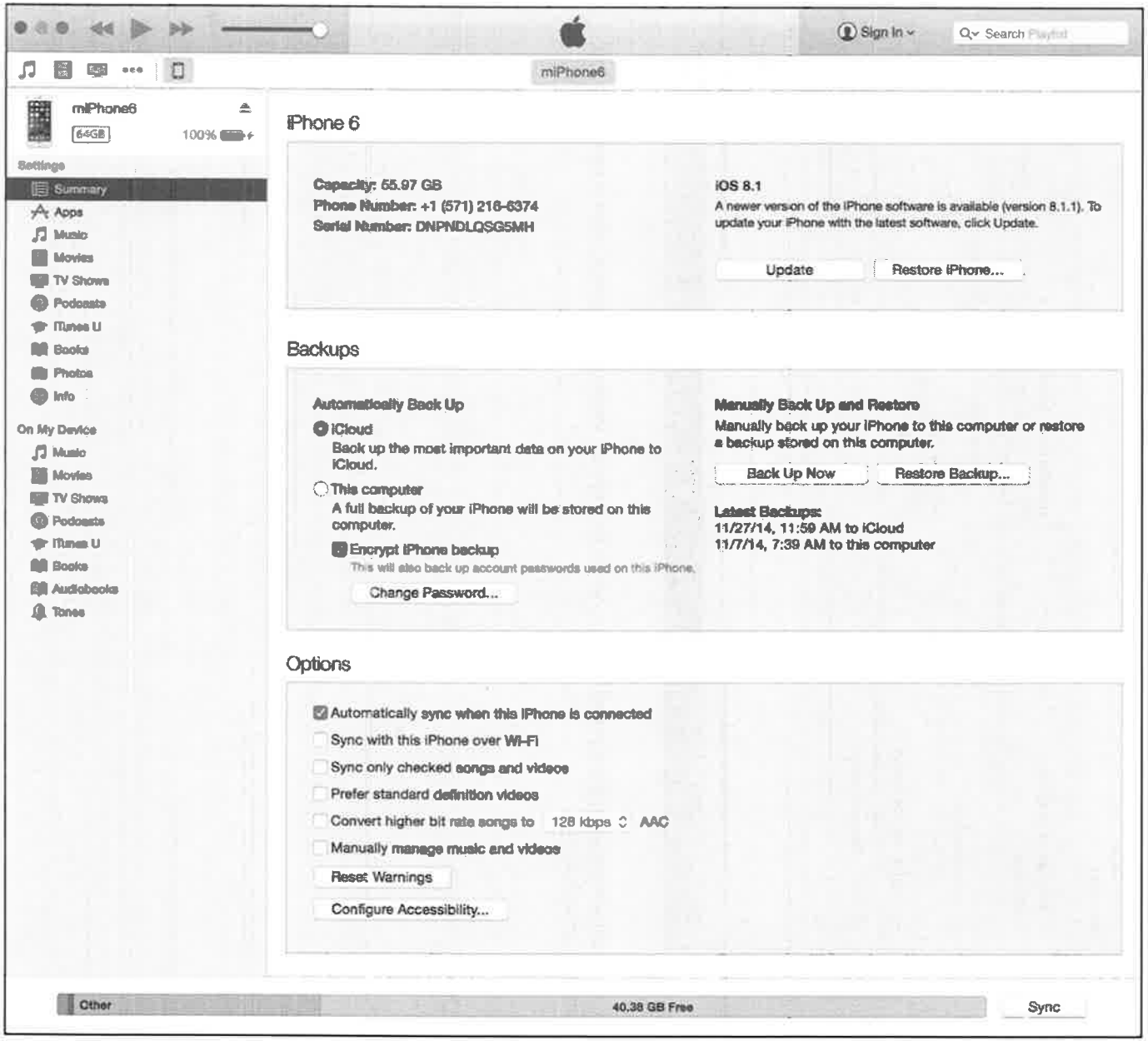
The `conn:128:Last Connect` key contains a hex representation of a Mac OS timestamp of the last device connection time in local system time.

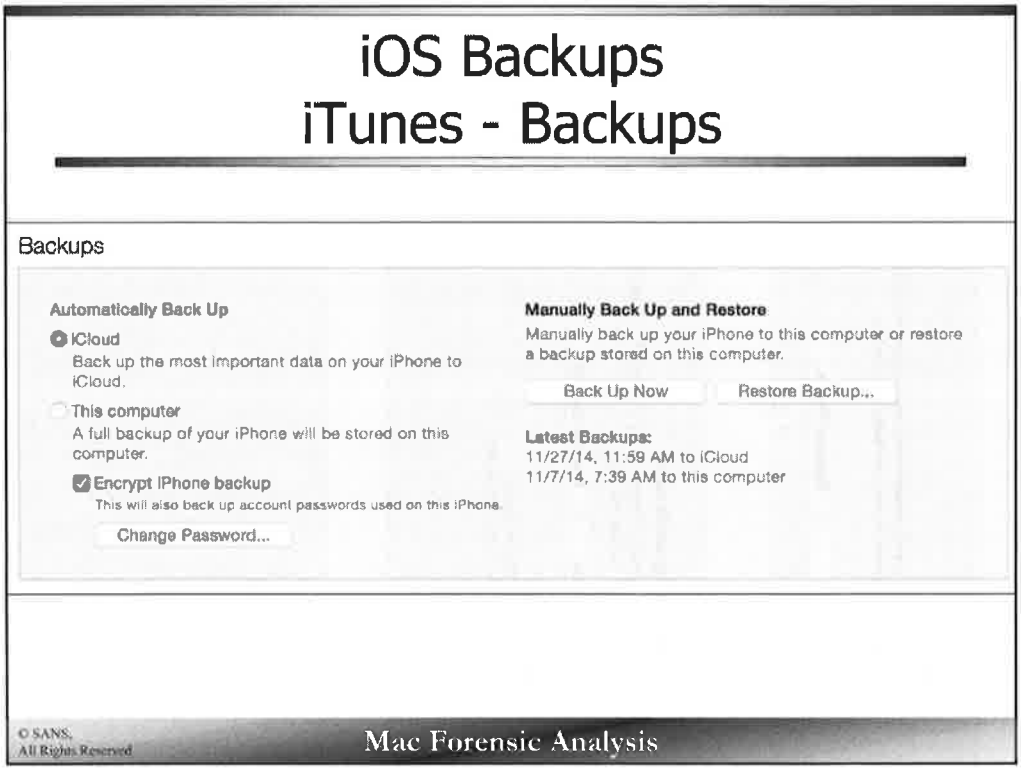
Key	Type	Value
▼ Root	Dictionary	(3 items)
▼ Devices	Dictionary	(3 items)
▶ A3626E4E1B1F010E	Dictionary	(11 items)
▼ F5C0E42BD971E840	Dictionary	(12 items)
Region Info	String	LL/A
Device Class	String	iPhone
IMEI	String	354409063007558
ID	String	F5C0E42BD971E840
Updater Family ID	Number	10,042
Serial Number	String	DNPNDLQSG5MH
Use Count	Number	2
Family ID	Number	10,042
Connected	Date	Nov 7, 2014, 2:52:17 PM
Firmware Version String	String	8.1
Firmware Version	Number	256
MEID	String	35440906300755
▼ ADD8A302AB39D8EA	Dictionary	(11 items)
Device Class	String	iPhone
ID	String	ADD8A302AB39D8EA
Use Count	Number	2
Region Info	String	LL/A
IMEI	String	012659001279644
Firmware Version String	String	6.1.6
Updater Family ID	Number	10,004
Family ID	Number	10,004
Firmware Version	Number	256
Serial Number	String	851174G1EDG
Connected	Date	Nov 29, 2014, 11:20:43 AM
com.apple.PreferenceSync.ExcludeAllSyncKeys	Boolean	YES
conn:128:Last Connect	Data	<d09f5c8b>

When a new iDevice connection is detected in iTunes, a new "tab" will be available that looks like an iPhone/iPad/iPod icon. This view gives us basic information including the type of devices, its capacity, name, iOS version (and updates available) and identifying information such as phone number (if available) and serial number.

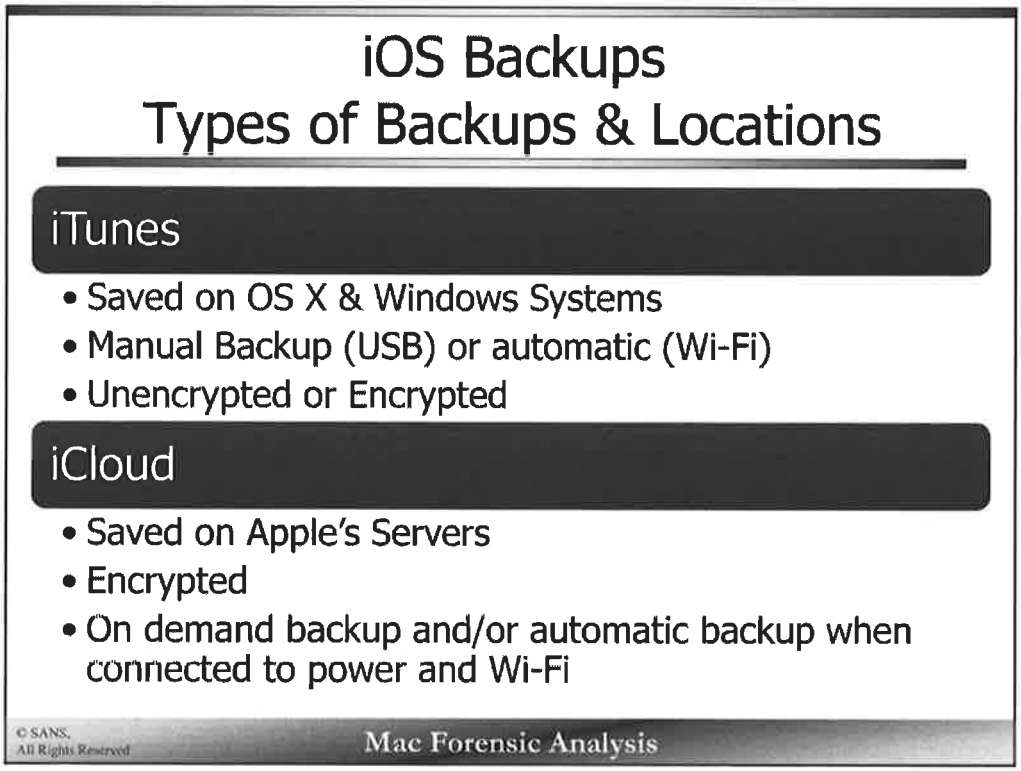
The Backups section allows a user to select whether they want to backup their device using iCloud, iTunes or both. Other information relating to previous backups or encrypted may also be available.

The Options section allows a user to select other syncing options such as whether to sync over Wi-Fi, automatically sync, and what to sync.


Sign In
Search Playlist
miPhone6
miPhone6
64GB
100%
Settings
Summary
Apps
Music
Movies
TV Shows
Podcasts
iTunes U
Books
Photos
Info
On My Device
Music
Movies
TV Shows
Podcasts
iTunes U
Books
Audiobooks
Tones
iPhone 6
Capacity: 55.97 GB
Phone Number: +1 (571) 216-6374
Serial Number: DNPNDLQSG5MH
iOS 8.1
A newer version of the iPhone software is available (version 8.1.1). To update your iPhone with the latest software, click Update.
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 also back up account passwords used on this iPhone.
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 Backups:
11/27/14, 11:59 AM to iCloud
11/7/14, 7:39 AM to this computer
Options
Automatically sync when this iPhone is connected
Sync with this iPhone over Wi-Fi
Sync only checked songs and videos
Prefer standard definition videos
Convert higher bit rate songs to 128 kbps AAC
Manually manage music and videos
Reset Warnings
Configure Accessibility...
Other
40.38 GB Free
Sync



This section will focus on the types of backups, where backups are located on a system, what is backed up and how to analyze backups.



iDevices can be backed up with iTunes locally on Mac or Windows systems or in the iCloud.

On local systems the iTunes backups use the same scheme, just the `/MobileSync/Backup/` directory location is different. iTunes backups may be backed up manually (USB) or automatically (Wi-Fi) and may use encryption or not.

Mac: `~/Library/Application Support/MobileSync/Backup/`

Windows XP: `\Documents and Settings\<user>\Application Data\Apple Computer\MobileSync\Backup\`

Windows Vista+: `\Documents and Settings\Users\<user>\AppData\Roaming\Apple Computer\MobileSync\Backup\`

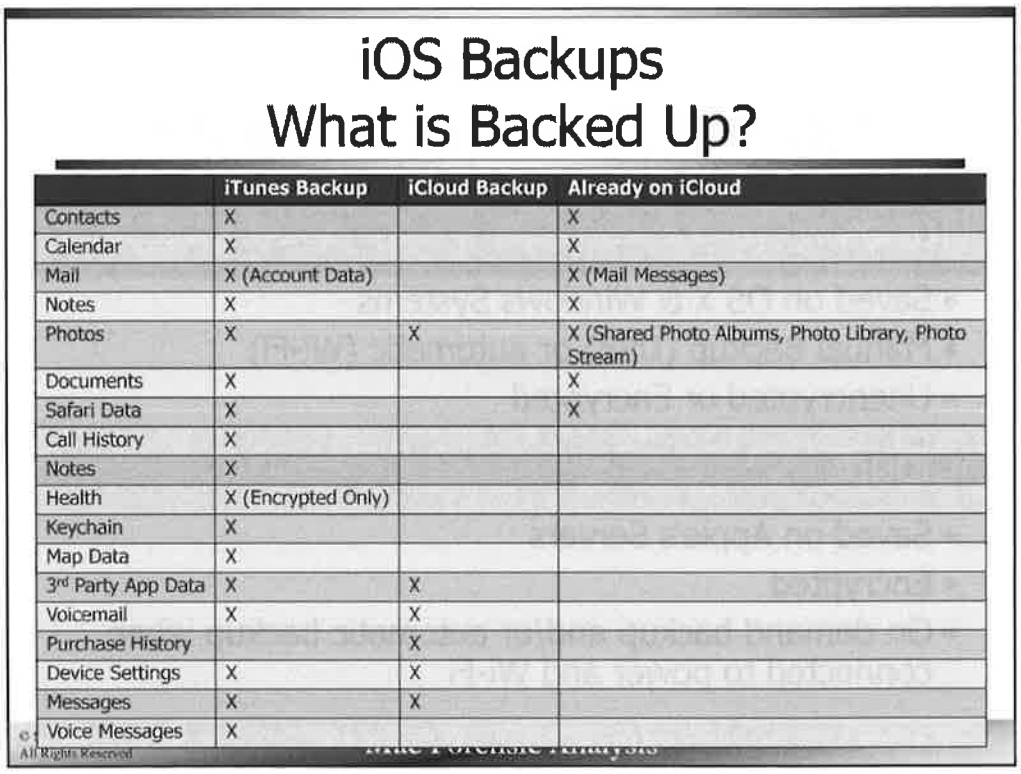
iCloud backups are stored on Apple's servers in an encrypted proprietary format. iCloud backups may be performed on demand or automatically but must be connected to power and Wi-Fi.

References:

http://support.apple.com/en-us/HT4946

http://support.apple.com/kb/ph12519

http://support.apple.com/en-us/HT5262



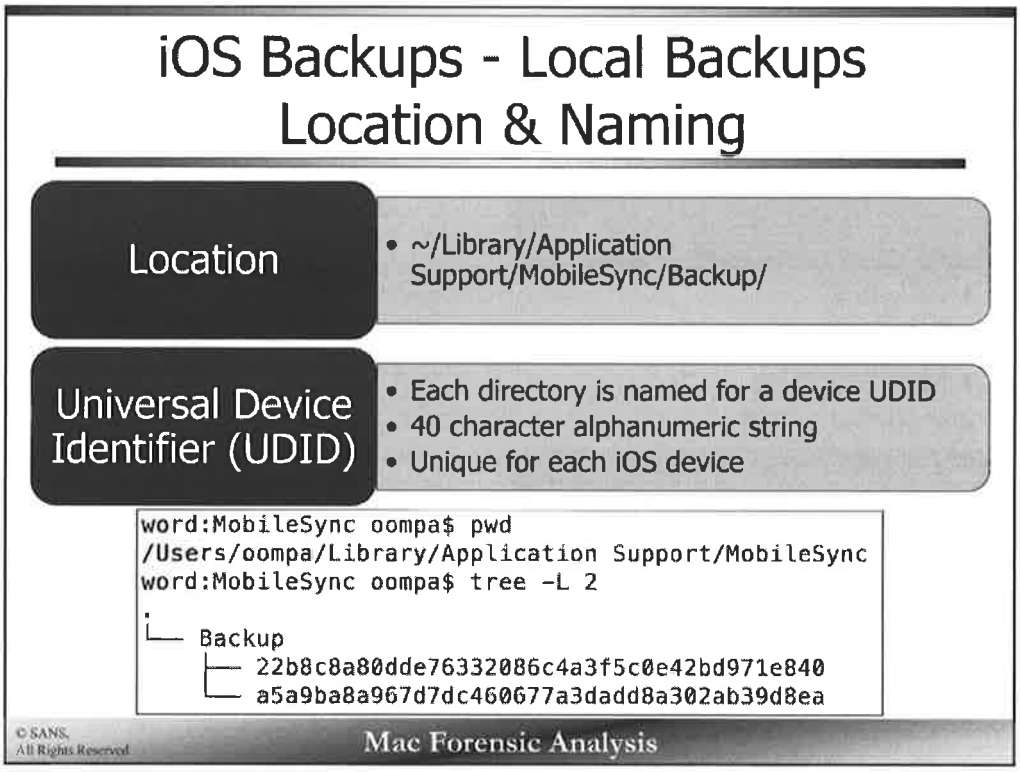
iTunes and iCloud backups may contain different items shown in the table above. If the user implements iCloud, data that is already synced with iCloud will not be backed up in their iCloud backups.

References:

http://support.apple.com/kb/PH12519

http://support.apple.com/en-us/HT4946

http://support.apple.com/en-us/HT5262

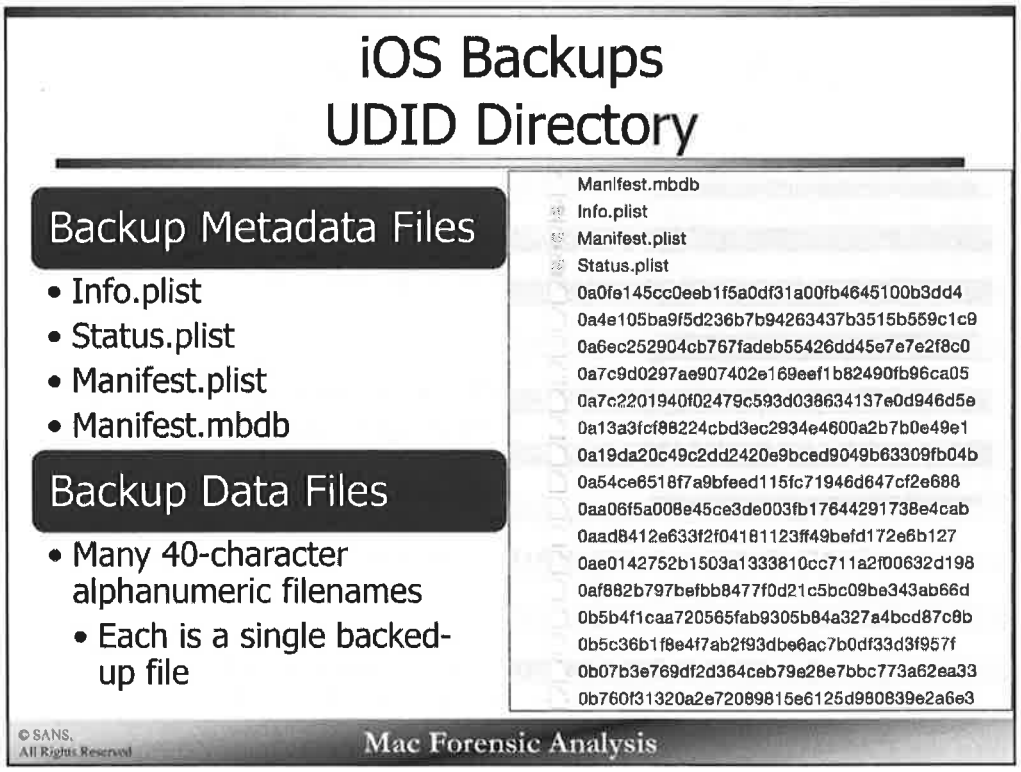


Local iTunes created backups are stored in a `/MobileSync/Backup/` directories located on the user's `Library/Application Support/` directory on OS X systems. This location differs slightly on Windows systems as show in a previous slide.

In the Backup directory, each device backed up will have a unique 40-character alphanumeric UDID directory containing the backup data for that device.

In the screenshot above two devices have been backed up under this user account, as there are two device UDID directories.

You may see directory names with a `UDID-<timestamp>`, these are created during a restore/update of the iDevice.

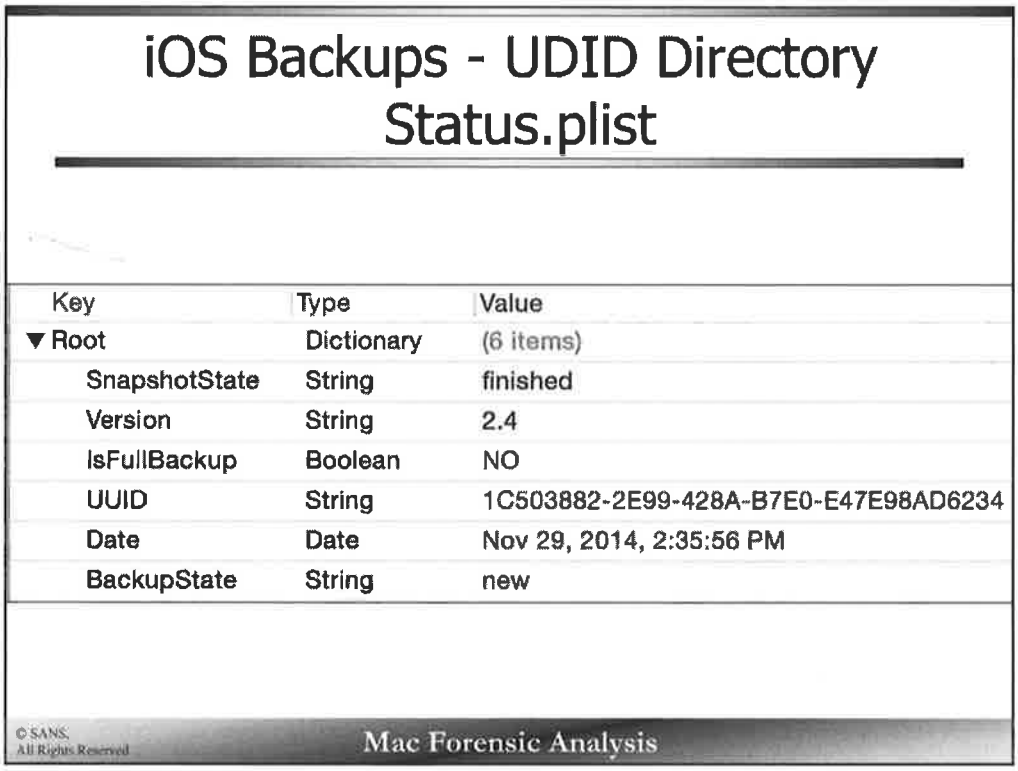


Each UDID named directory contains metadata and data files for a backed up device.

The following files contain backup metadata and will be described further in upcoming slides.

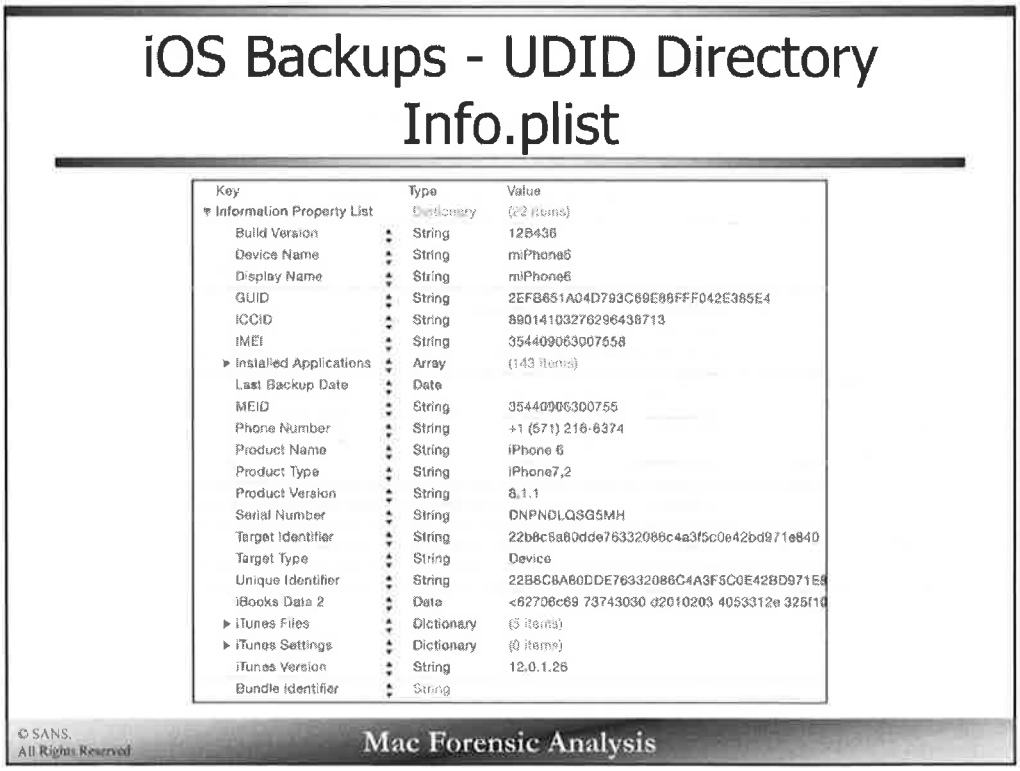
- `Info.plist`
- `Status.plist`
- `Manifest.plist`
- `Manifest.mbdb`

The remaining files are the backup data files, each file represents on file that has been backed up on the device. Each 40-character alphanumeric filename is the SHA1 hash of



The `Status.plist` file contains the following information:

- `SnapshotState` – Status of this backup snapshot
- `isFullBackup` – Is this a full backup or not
- `Date` – Timestamp of the backup
- `BackupState` – Type of Backup



The `Info.plist` file contains the following:

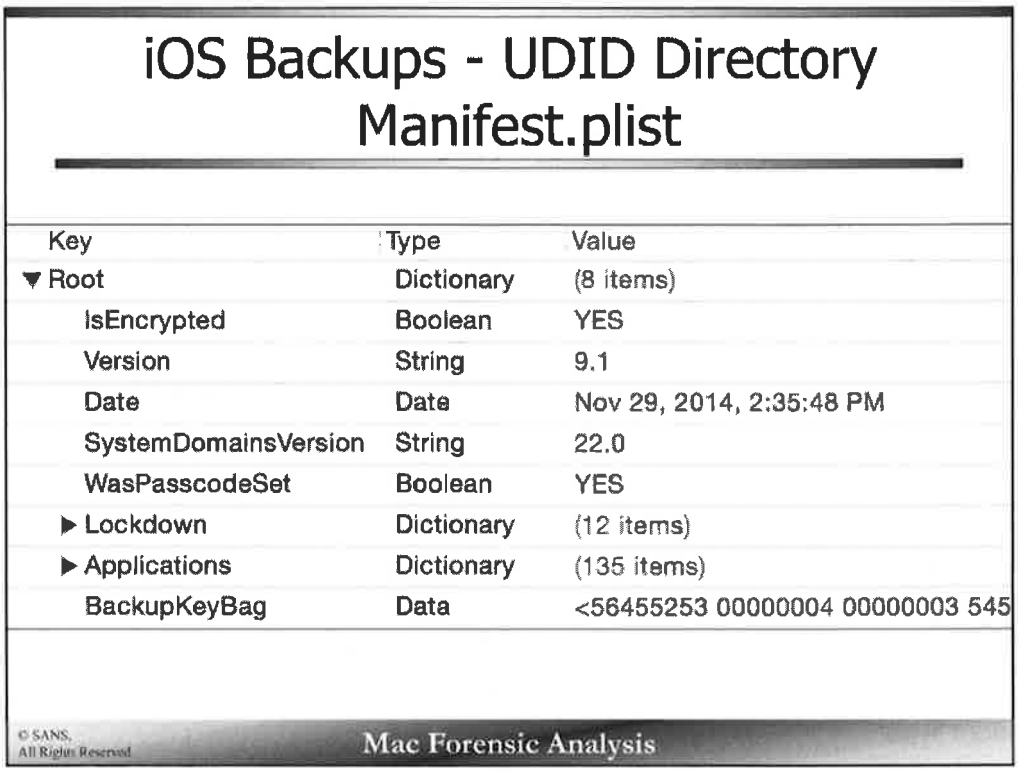
- Device Name
- Device Identifiers (GUID/ICCID/MEID/IMEI/Serial Number/UDID)
- Phone Number
- Make/Model/Build Data
- iOS Version
- Last Backup Date (The screen shot shows this blank – it appears to be a bug with Xcode, it is viewable using the `plutil` utility).
- Installed Applications – Contains the bundle identifiers for each application installed, including native iOS apps.

If the applications were synced with an OS X system you will see iPhone Application (IPA) files in the `~/Music/iTunes/iTunes Media/Mobile Applications/` directory. IPA files are ZIP archives of the iPhone applications.

Sometimes the `iTunes Settings/LibraryApplications/` key is populated with other applications that the user may have previously downloaded but does not currently have installed.

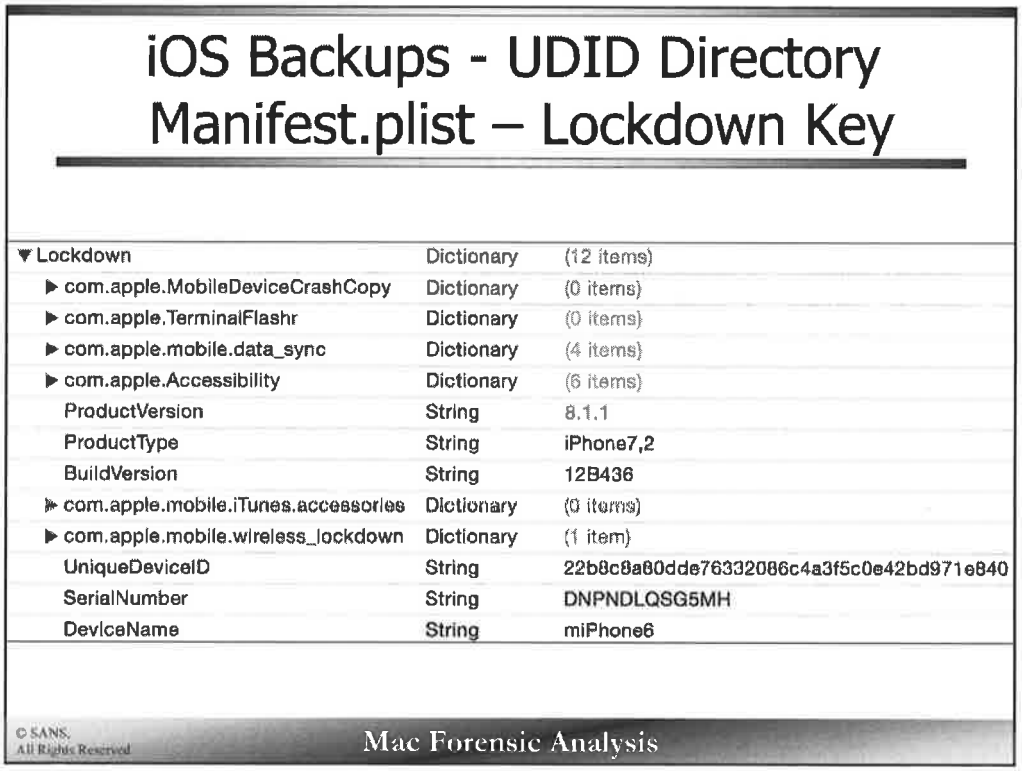
▼ iTunes Settings	Dictionary	(1 item)
▼ LibraryApplications	Array	(24 items)
Item 0	String	altsource.MyConsumerCellular
Item 1	String	cmmjhd01
Item 2	String	com.5thfinger.redshop.joann
Item 3	String	com.amazon.Lassen
Item 4	String	com.apple.iBooks
Item 5	String	com.chillingo.cuttherope
Item 6	String	com.clickgamer.AngryBirds

Key	Type	Value
▼ Information Property List	Dictionary	(22 items)
Build Version	String	12B436
Device Name	String	miPhone6
Display Name	String	miPhone6
GUID	String	2EFB651A04D793C69E86FFF042E385E4
ICCID	String	89014103276296438713
IMEI	String	354409063007558
▶ Installed Applications	Array	(143 items)
Last Backup Date	Date	
MEID	String	35440906300755
Phone Number	String	+1 (571) 216-6374
Product Name	String	iPhone 6
Product Type	String	iPhone7,2
Product Version	String	8.1.1
Serial Number	String	DNPNDLQSG5MH
Target Identifier	String	22b8c8a80dde76332086c4a3f5c0e42bd971e840
Target Type	String	Device
Unique Identifier	String	22B8C8A80DDE76332086C4A3F5C0E42BD971E8
iBooks Data 2	Data	<62706c69 73743030 d2010203 4053312e 325f10
▶ iTunes Files	Dictionary	(5 items)
▶ iTunes Settings	Dictionary	(0 items)
iTunes Version	String	12.0.1.26
Bundle Identifier	String	



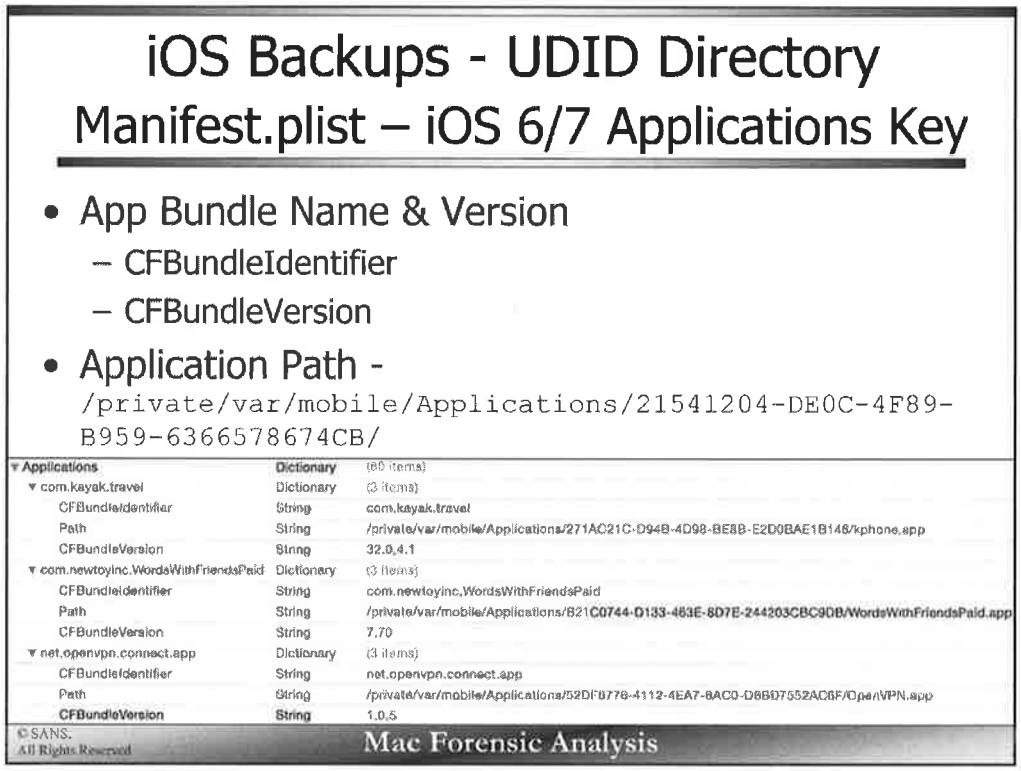
The `Manifest.plist` file contains the following information:

- `isEncrypted` – Is the backup encrypted or not
- `Date` – When was the backup created
- `WasPasscodeSet` – Was the passcode set on the device



The `Manifest.plist` file also contains the `Lockdown` key which contains device identification information such as name, serial number, UDID. It also contains version information for the installed iOS, device make/model, and build data.

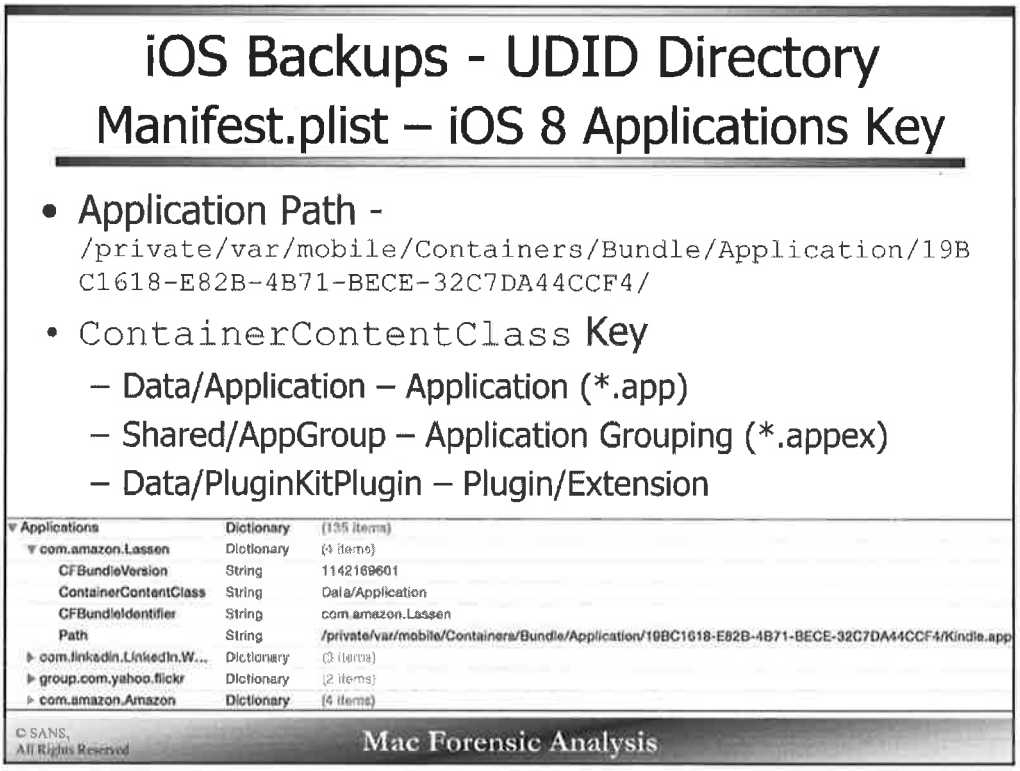
▼ Lockdown	Dictionary	(12 items)
▶ com.apple.MobileDeviceCrashCopy	Dictionary	(0 items)
▶ com.apple.TerminalFlashr	Dictionary	(0 items)
▶ com.apple.mobile.data_sync	Dictionary	(4 items)
▶ com.apple.Accessibility	Dictionary	(6 items)
ProductVersion	String	8.1.1
ProductType	String	iPhone7,2
BuildVersion	String	12B436
▶ com.apple.mobile.iTunes.accessories	Dictionary	(0 items)
▶ com.apple.mobile.wireless_lockdown	Dictionary	(1 item)
UniqueDeviceID	String	22b8c8a80dde76332086c4a3f5c0e42bd971e840
SerialNumber	String	DNPNDLQSG5MH
DeviceName	String	miPhone6



The `Manifest.plist` file also keeps track of what applications were installed on the device in the Applications key.

Each item under this key contains the bundle identifier for the application (usually in reverse DNS format). Under each application bundle ID, contains information about the application including where its files are stored on the device. In the screenshot above the Kindle applications files are stored in `/private/var/mobile/Containers/Bundle/Application/19BC1618-E82B-4B71-BECE-32C7DA44CCF4/`

▼ Applications	Dictionary	(80 items)
▼ com.kayak.travel	Dictionary	(3 items)
CFBundleIdentifier	String	com.kayak.travel
Path	String	/private/var/mobile/Applications/271AC21C-D94B-4D98-BE8B-E2D0BAE1B148/kphone.app
CFBundleVersion	String	32.0.4.1
▼ com.newtoyinc.WordsWithFriendsPaid	Dictionary	(3 items)
CFBundleIdentifier	String	com.newtoyinc.WordsWithFriendsPaid
Path	String	/private/var/mobile/Applications/B21C0744-D133-463E-8D7E-244203CBC9DB/WordsWithFriendsPaid.app
CFBundleVersion	String	7.70
▼ net.openvpn.connect.app	Dictionary	(3 items)
CFBundleIdentifier	String	net.openvpn.connect.app
Path	String	/private/var/mobile/Applications/52DF8776-4112-4EA7-8AC0-D6BD7552AC6F/OpenVPN.app
CFBundleVersion	String	1.0.5



The `Manifest.plist` file also keeps track of what applications were installed on the device in the Applications key.

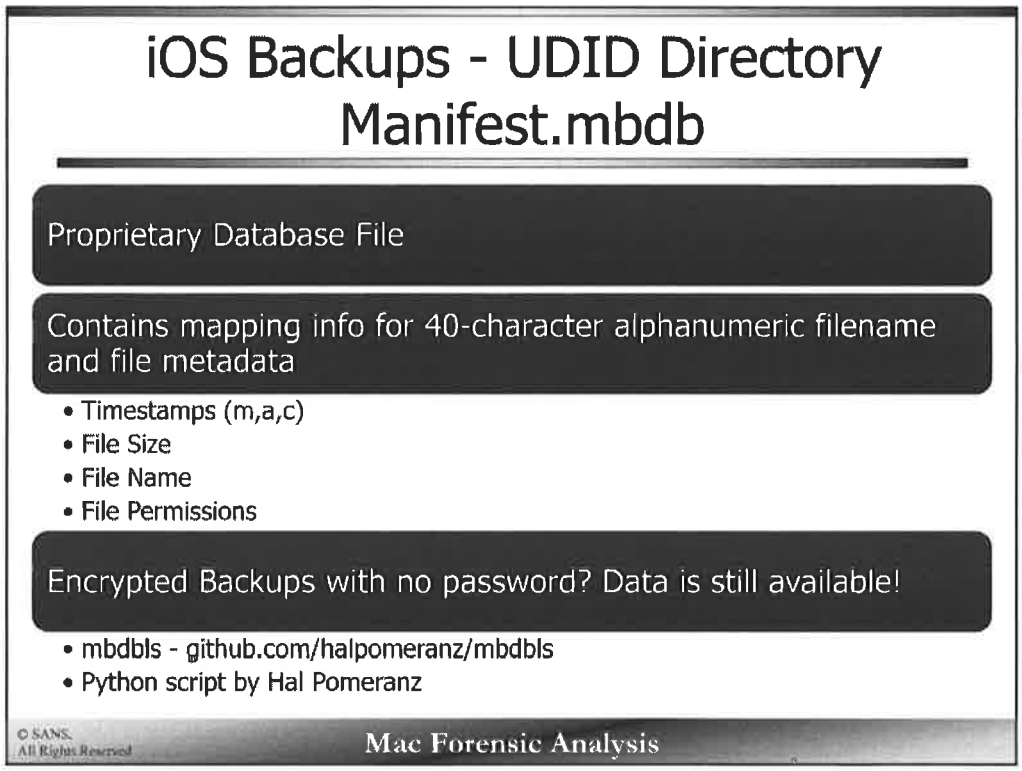
Each item under this key contains the bundle identifier for the application (usually in reverse DNS format). Under each application bundle ID, contains information about the application including where its files are stored on the device. In the screenshot above the Kindle applications files are stored in `/private/var/mobile/Containers/Bundle/Application/19BC1618-E82B-4B71-BECE-32C7DA44CCF4/`

iOS 8 has introduced Containers to iOS which include a new key, `ContainerContentClass` – which shows if this is an application, extension, or an Application Group.

Other keys present (unless it is an App Group) will show application bundle name and version information:

- `CFBundleVersion` – Application Version
- `CFBundleIdentifier` – Application Bundle

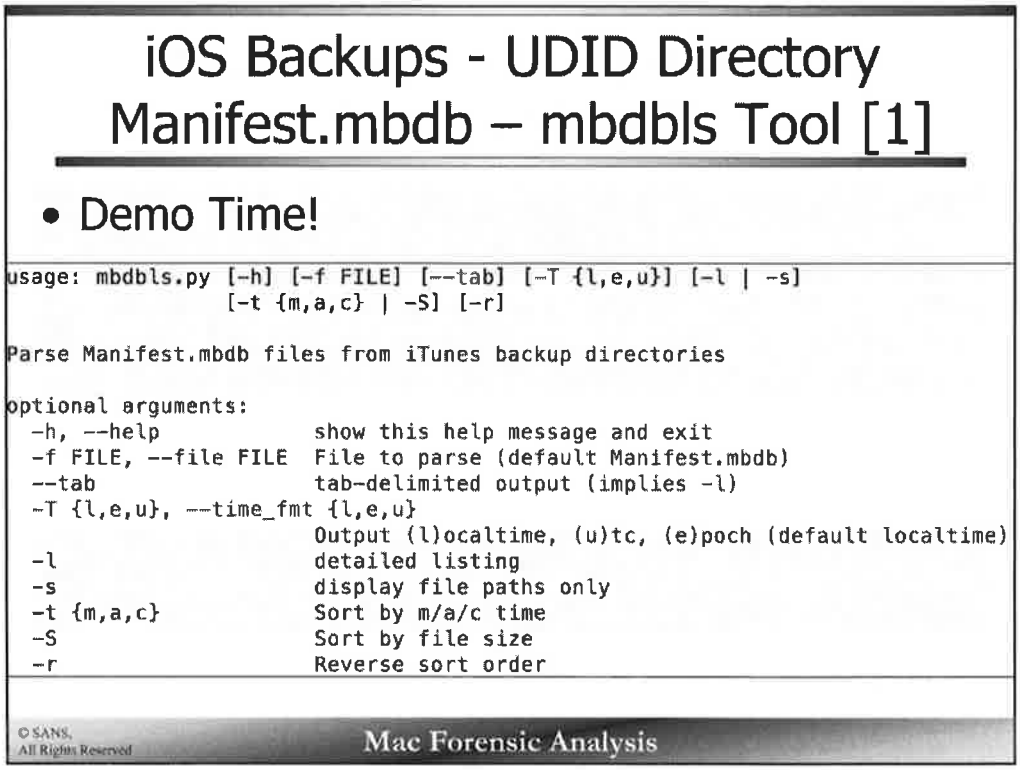
▼ Applications	Dictionary	(135 items)
▼ com.amazon.Lassen	Dictionary	(4 items)
CFBundleVersion	String	1142169601
ContainerContentClass	String	Data/Application
CFBundleIdentifier	String	com.amazon.Lassen
Path	String	/private/var/mobile/Containers/Bundle/Application/19BC1618-E82B-4B71-BECE-32C7DA44CCF4/Kindle.app
▶ com.linkedin.LinkedIn.W...	Dictionary	(3 items)
▶ group.com.yahoo.flickr	Dictionary	(2 items)
▶ com.amazon.Amazon	Dictionary	(4 items)



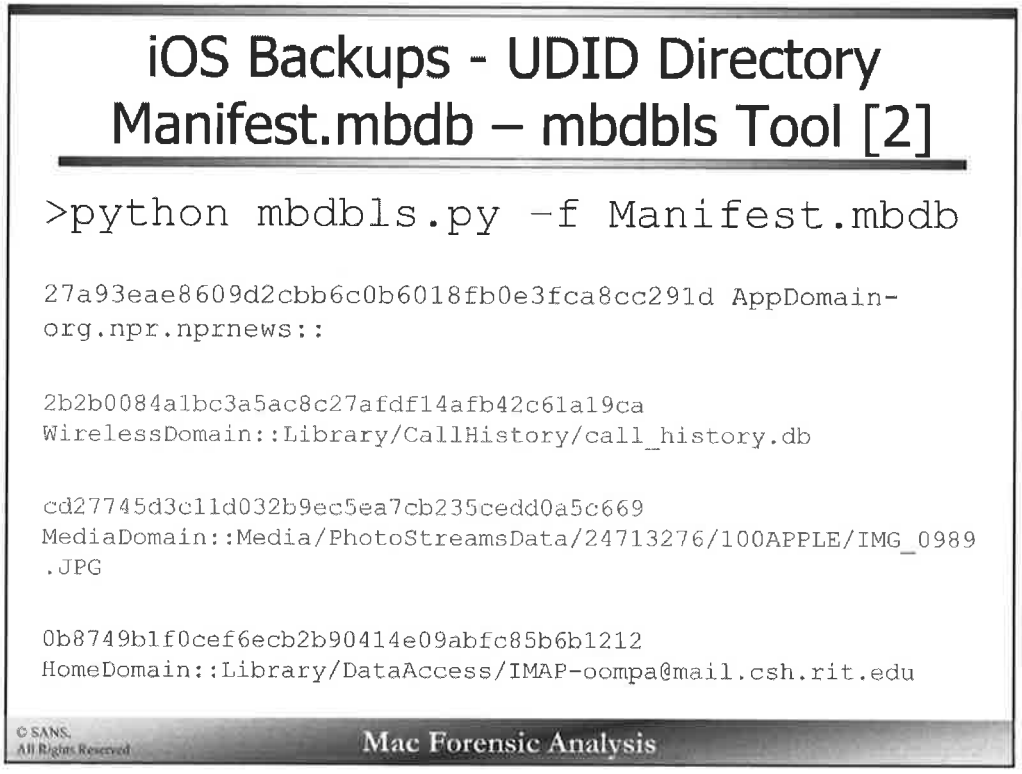
The `manifest.mbdb` file is a proprietary database file that contains the mapping for the 40-character named data files to their original location and metadata on the device.

The metadata includes everything from timestamps to file size, to filenames.

This database is useful in cases where an investigator does not have a password or is unable to brute force a password for an encrypted backup. You can still show evidence is a specific app on a device, or an e-mail account used.



This page intentionally left blank.



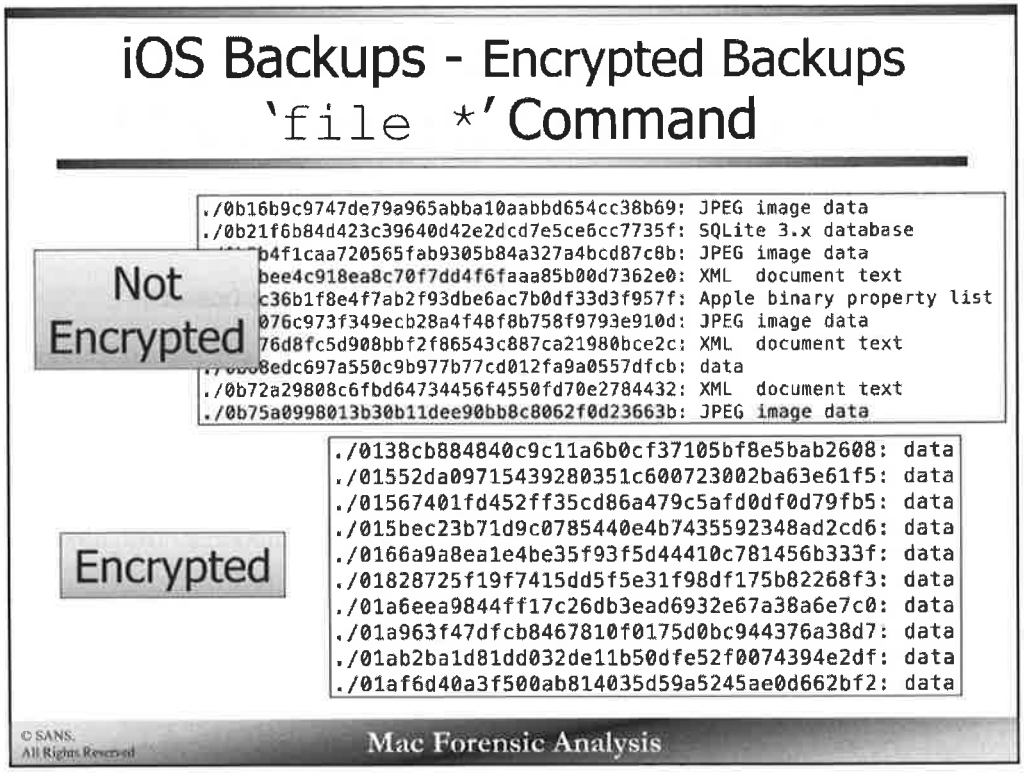
Each 40-character alphanumeric filename has an associated Domain and File Path. These are SHA1 hashed to create this filename.

<Domain>-<FilePath> = SHA1 filename, Try this out using http://www.sha1-online.com

For example, the SHA1 hash of `WirelessDomain-Library/CallHistory/call_history.db` is `2b2b0084a1bc3a5ac8c27afdf14afb42c61a19ca`

Each file in a backup as an associated domain this can be one of the following depending on the type of file:

- `BooksDomain`
- `CameraRollDomain`
- `DatabaseDomain`
- `HomeDomain`
- `KeychainDomain`
- `ManagedPreferencesDomain`
- `MediaDomain`
- `MobileDeviceDomain`
- `RootDomain`
- `SystemPreferencesDomain`
- `TonesDomain`
- `WirelessDomain`
- `AppDomain`

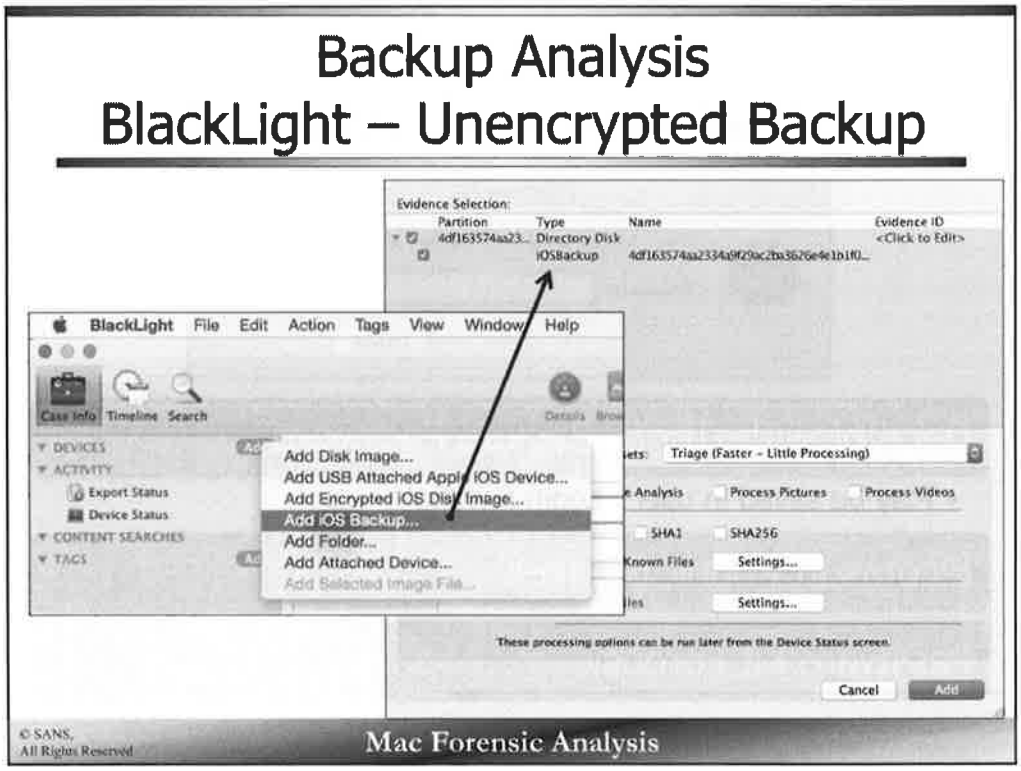


To quickly tell if a backup is encrypted, you can use the file utility built into OS X to determine the contents of the backed up files.

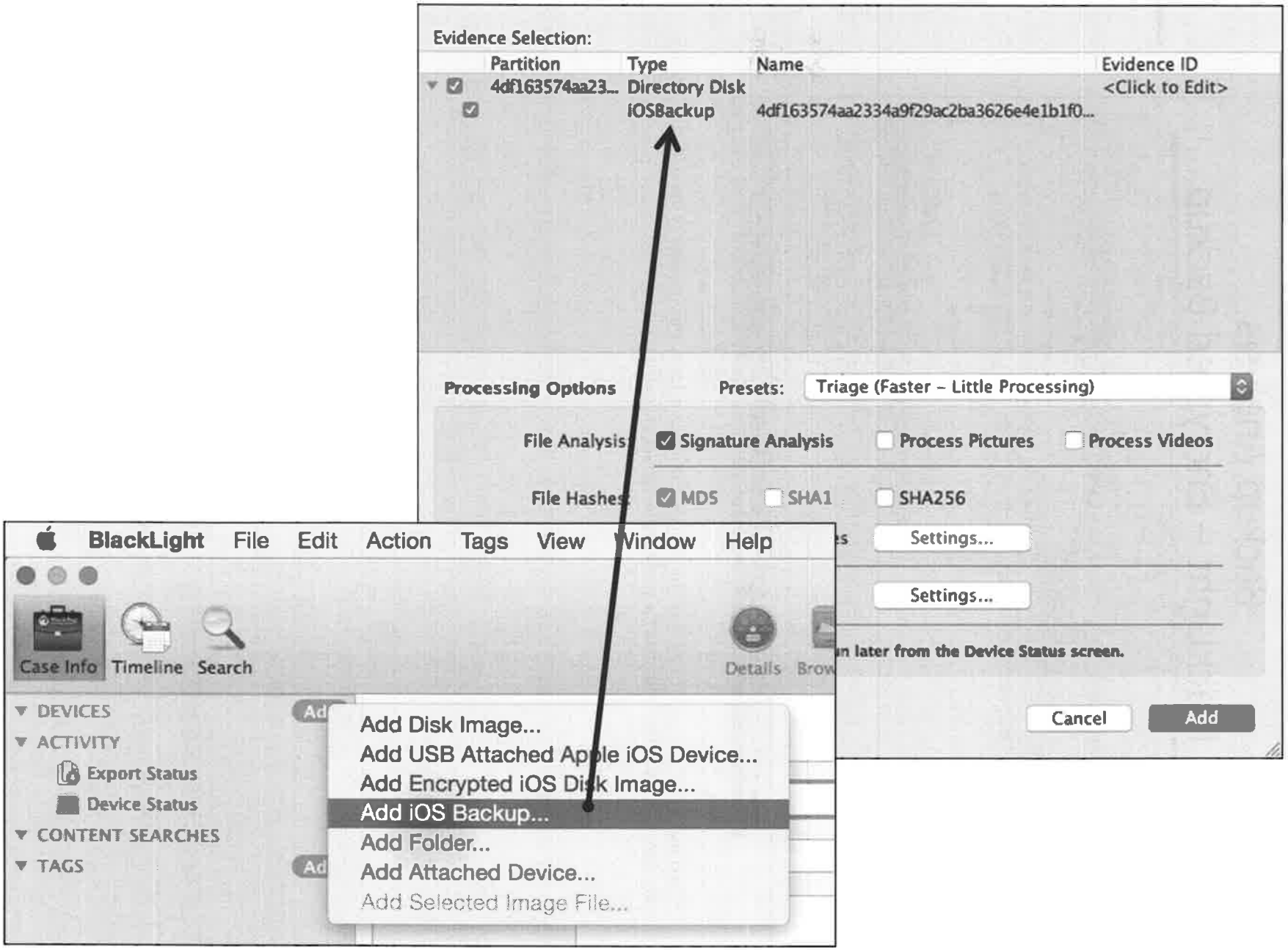
If they show the generic "`data`", they are likely encrypted, if they show property lists, pictures, and SQLite databases – it is not encrypted!

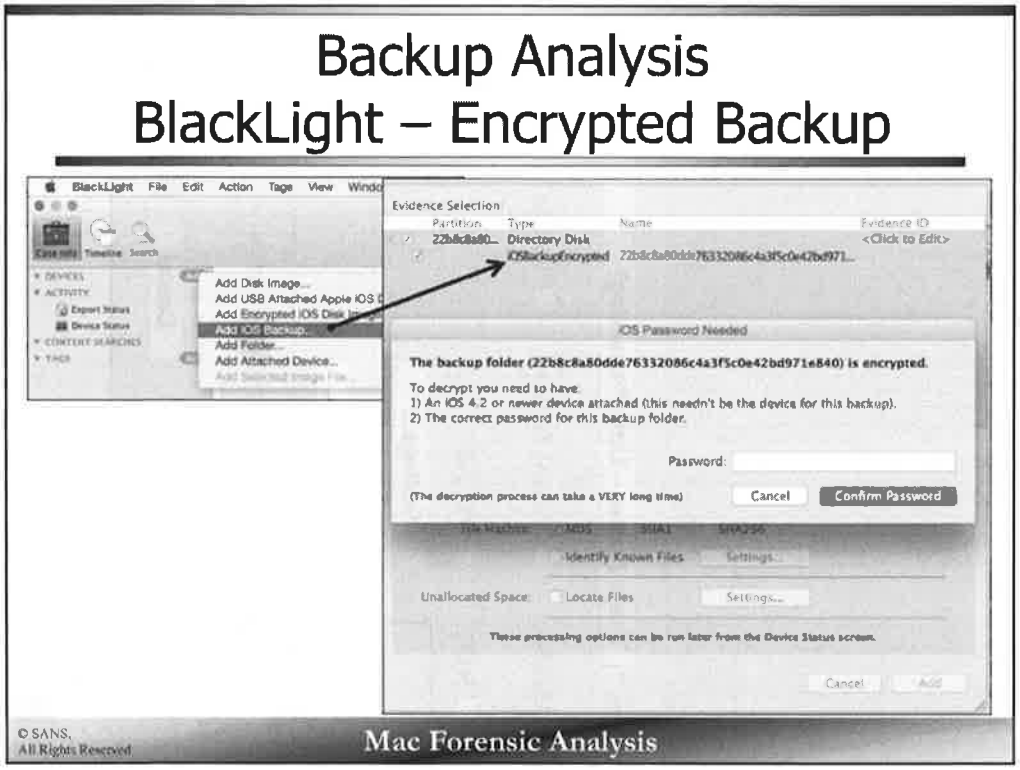
If you are able to view the users login keychain information, the backup password may be available is the user chose to save it in their keychain.

If not, you may attempt to crack an iOS backup password with various tools. Two tools that are currently available are Elcomsoft Phone Breaker (previous Elcomsoft Phone Password Breaker) and Passware.

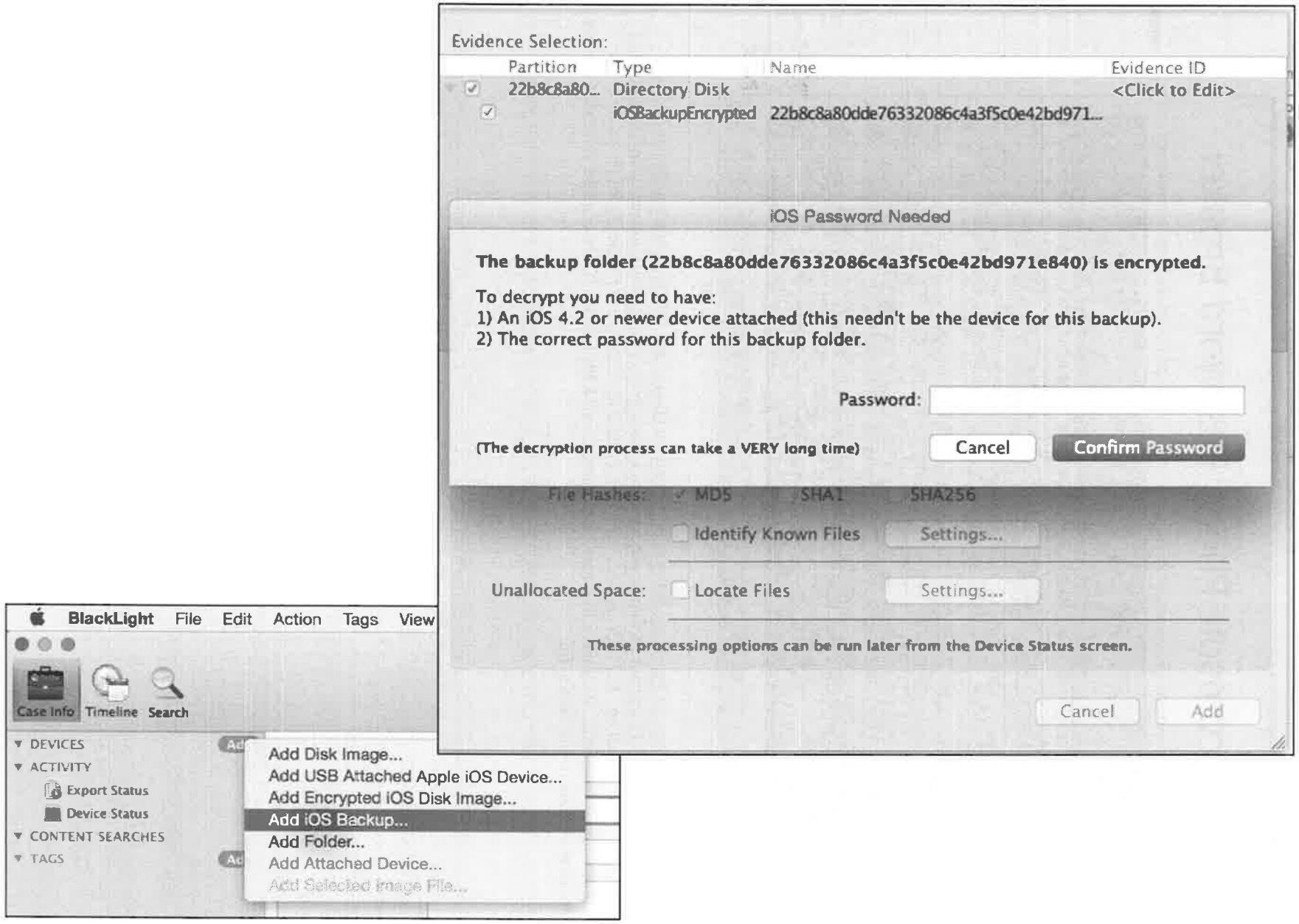


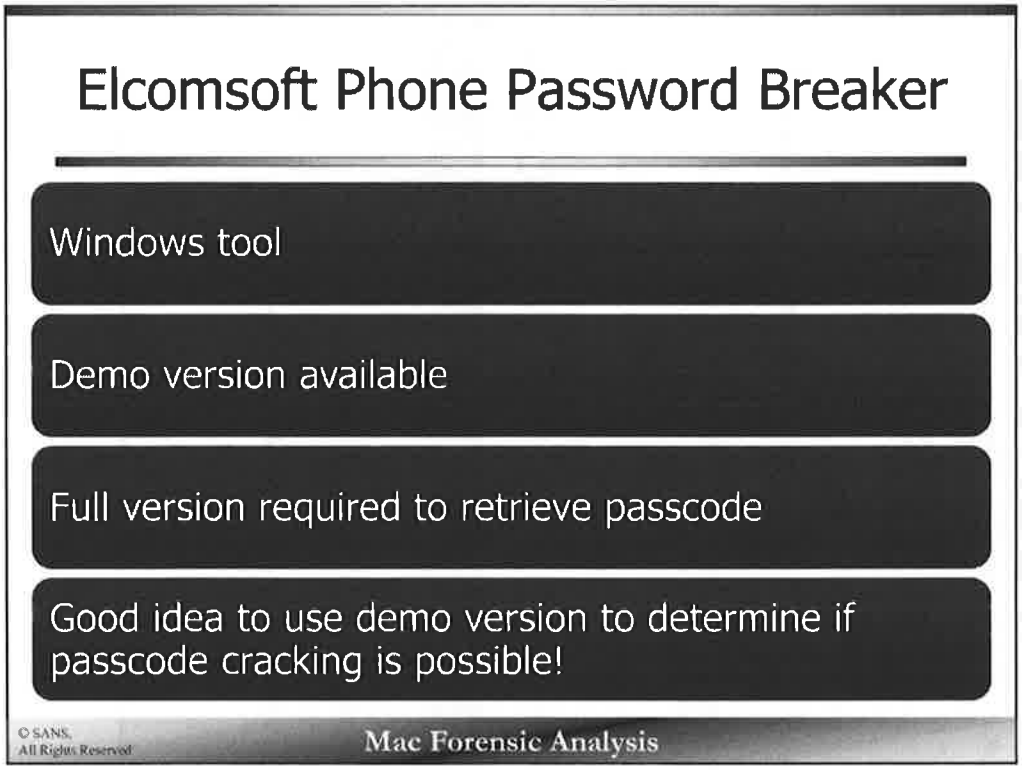
In Blacklight, you may add an iOS backup using the "`Add iOS Backup...`" option in the "Add" menu. This will open a window that shows the type of backup, its UDID, and whether or not it is encrypted.


Evidence Selection:
Partition
Type
Name
Evidence ID
4df163574aa23...
Directory Disk
<Click to Edit>
iOSBackup
4df163574aa2334a9f29ac2ba3626e4e1b1f0...
Processing Options
Presets:
Triage (Faster – Little Processing)
File Analysis
Signature Analysis
Process Pictures
Process Videos
File Hashes
MD5
SHA1
SHA256
Settings...
Settings...
later from the Device Status screen.
Cancel
Add
BlackLight
File
Edit
Action
Tags
View
Window
Help
Case Info
Timeline
Search
Details
DEVICES
ACTIVITY
Export Status
Device Status
CONTENT SEARCHES
TAGS
Add Disk Image...
Add USB Attached Apple iOS Device...
Add Encrypted iOS Disk Image...
Add iOS Backup...
Add Folder...
Add Attached Device...
Add Selected Image File...



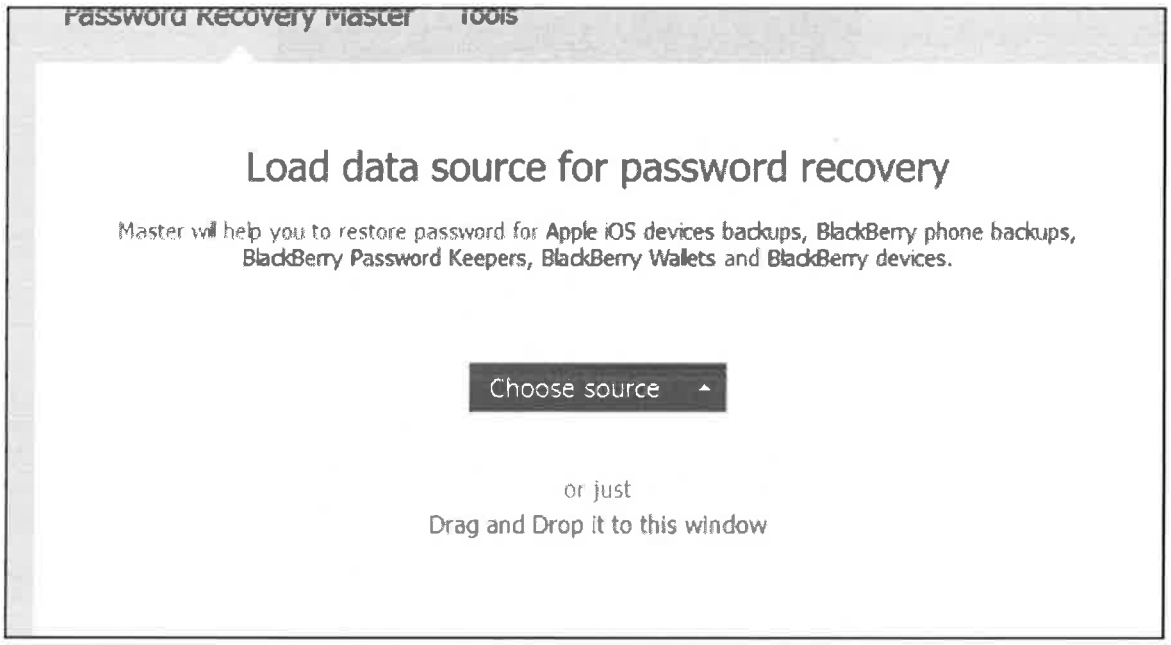
If you see "`iOSBackupEncrypted`" you will have to attach a device (not necessarily the evidentiary device, any device) to the system and input the password to decrypt the backup.

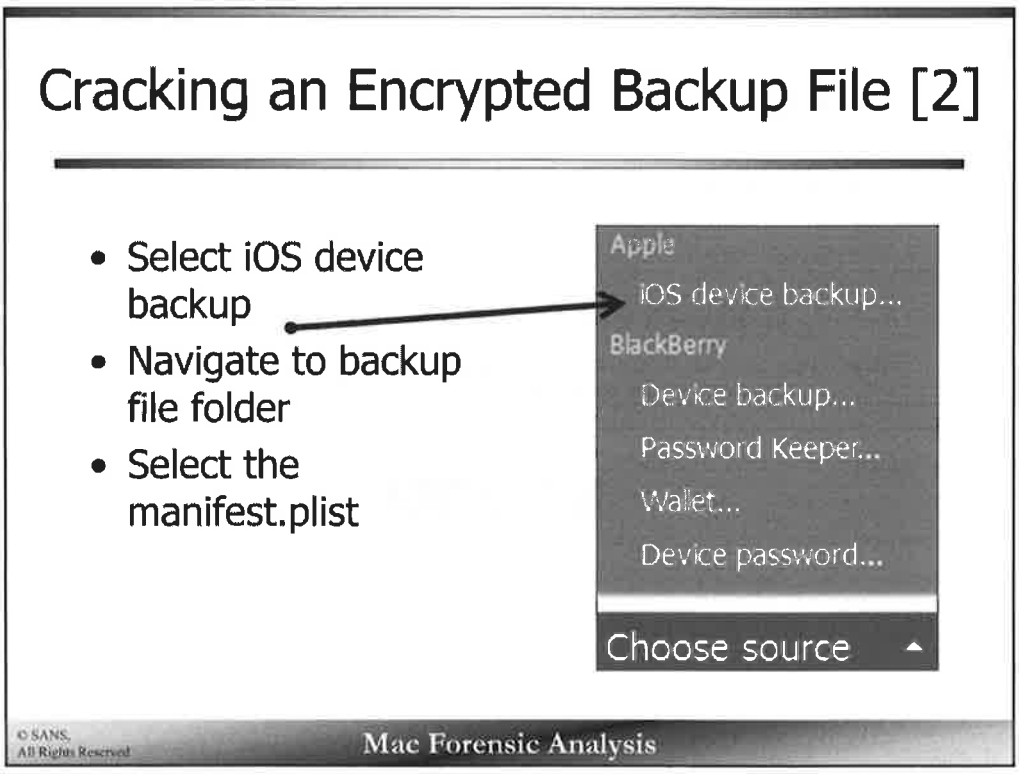

Evidence Selection:
Partition
Type
Name
Evidence ID
22b8c8a80...
Directory Disk
<Click to Edit>
iOSBackupEncrypted
22b8c8a80dde76332086c4a3f5c0e42bd971...
iOS Password Needed
The backup folder (22b8c8a80dde76332086c4a3f5c0e42bd971e840) is encrypted.
To decrypt you need to have:
1) An iOS 4.2 or newer device attached (this needn't be the device for this backup).
2) The correct password for this backup folder.
Password:
(The decryption process can take a VERY long time)
Cancel
Confirm Password
File Hashes:
MD5
SHA1
SHA256
Identify Known Files
Settings...
Unallocated Space:
Locate Files
Settings...
These processing options can be run later from the Device Status screen.
Cancel
Add
BlackLight
File
Edit
Action
Tags
View
Case Info
Timeline
Search
DEVICES
ACTIVITY
Export Status
Device Status
CONTENT SEARCHES
TAGS
Add Disk Image...
Add USB Attached Apple iOS Device...
Add Encrypted iOS Disk Image...
Add iOS Backup...
Add Folder...
Add Attached Device...
Add Selected Image File...



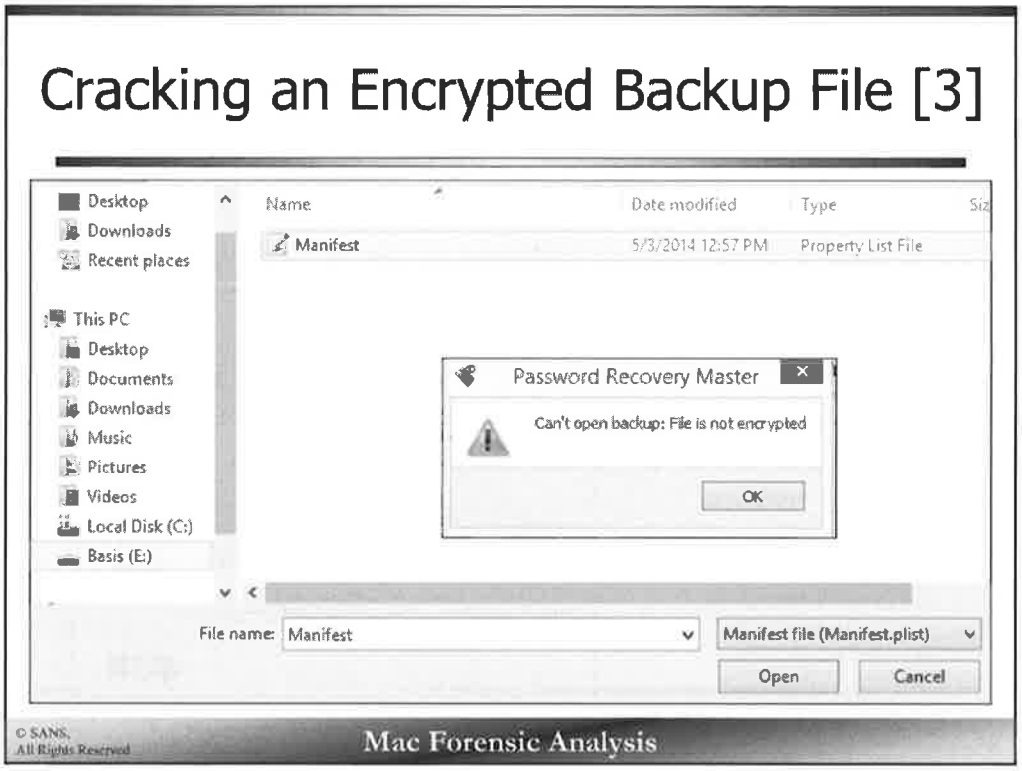
Elcomsoft Phone Password Breaker (EPPB) as previously mentioned is a Windows tool capable of cracking locked backup files. This feature will not work on iCloud backups as the Apple ID and password is required to access any iCloud backup. For backups set with a PIN, brute force attacks can be successful in seconds. For more complex passcodes, dictionary files may be required to crack the backup. Once cracked, forensic tools can be used to examine the backup file.

EPPB can crack encrypted backup files from iOS and BlackBerry devices. To select your backup file, drag and drop the backup folder into the window or select "Choose source" and navigate to your file.

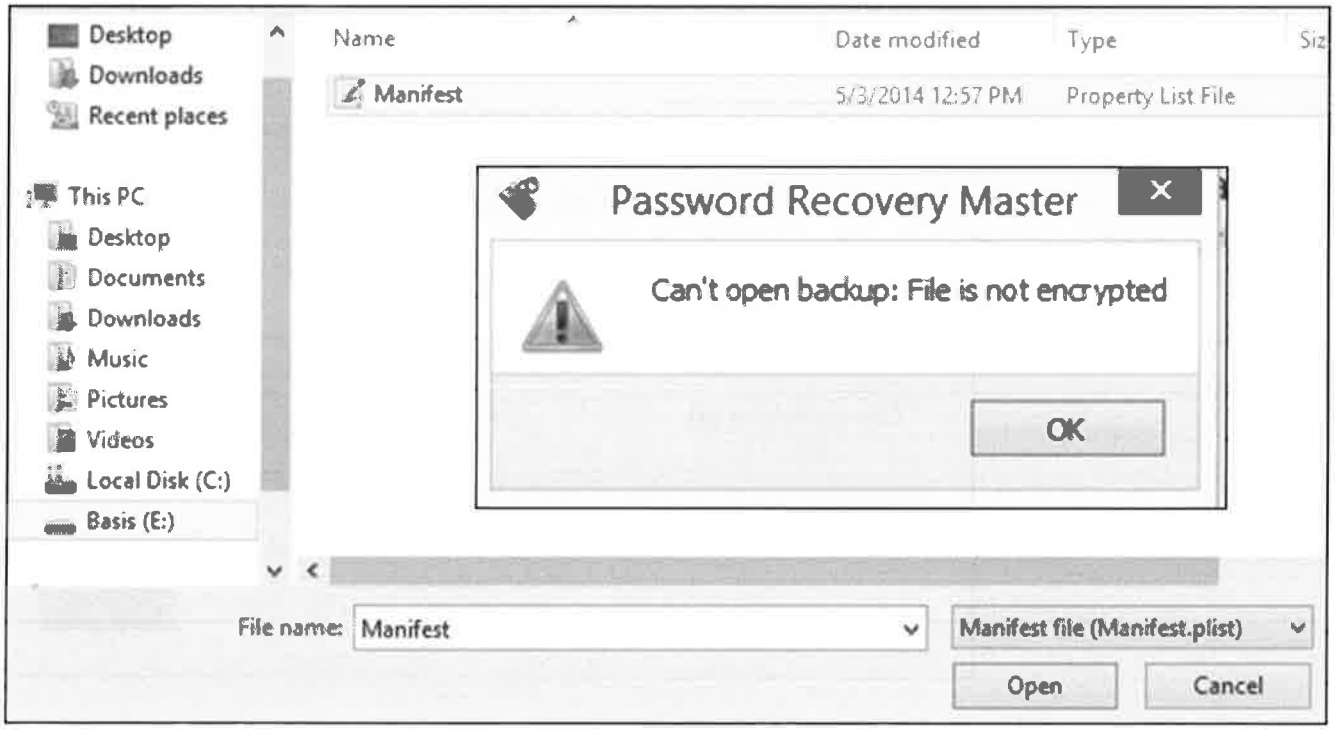


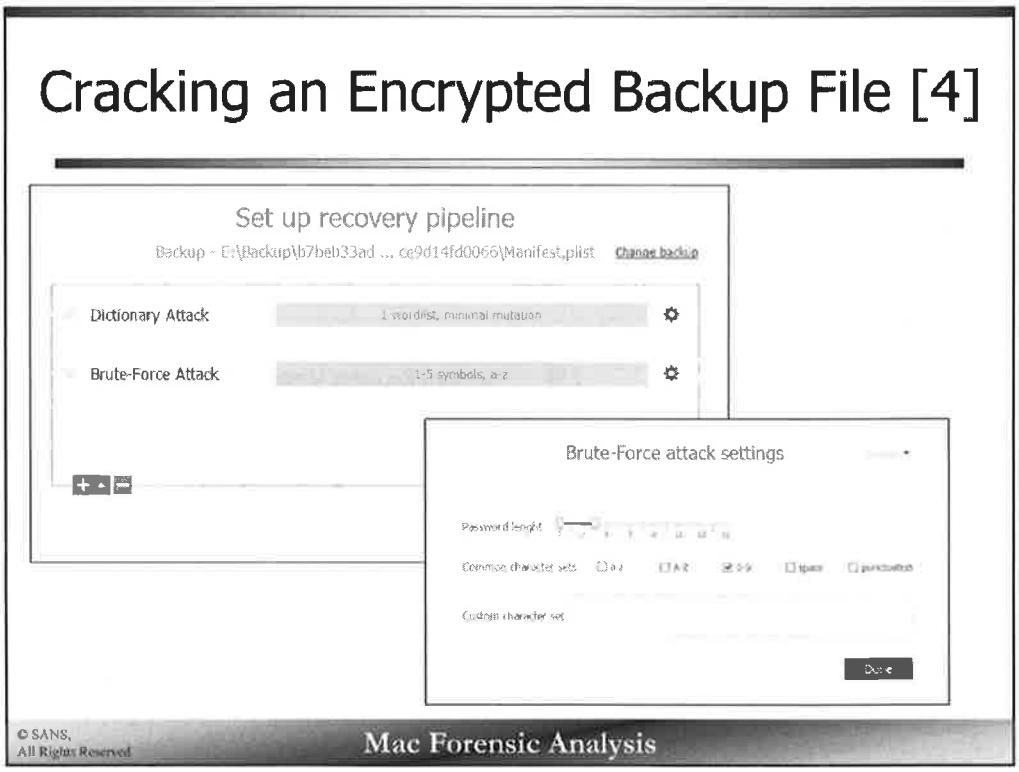


First select iOS device backup and then navigate to the backup file folder. The manifest.plist file contains the encryption status flag, which is required by the tool.

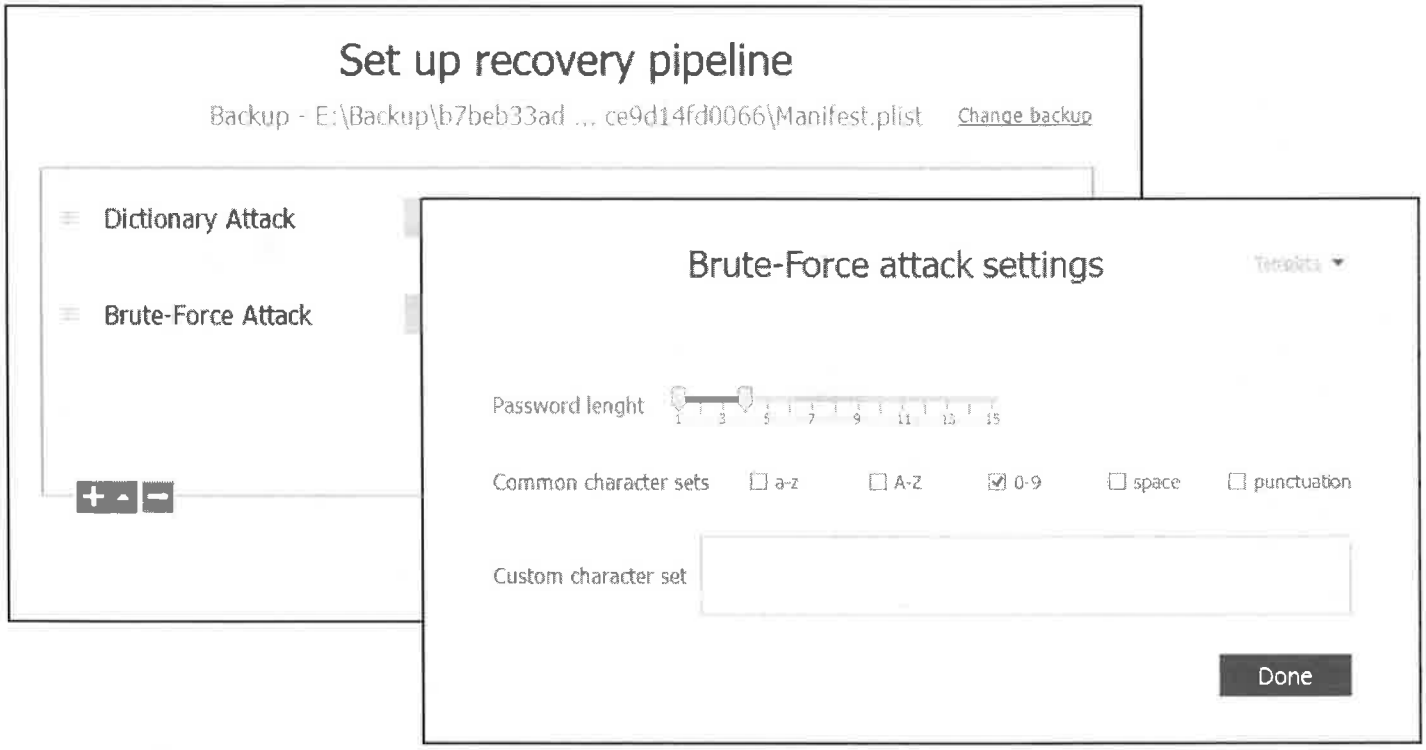


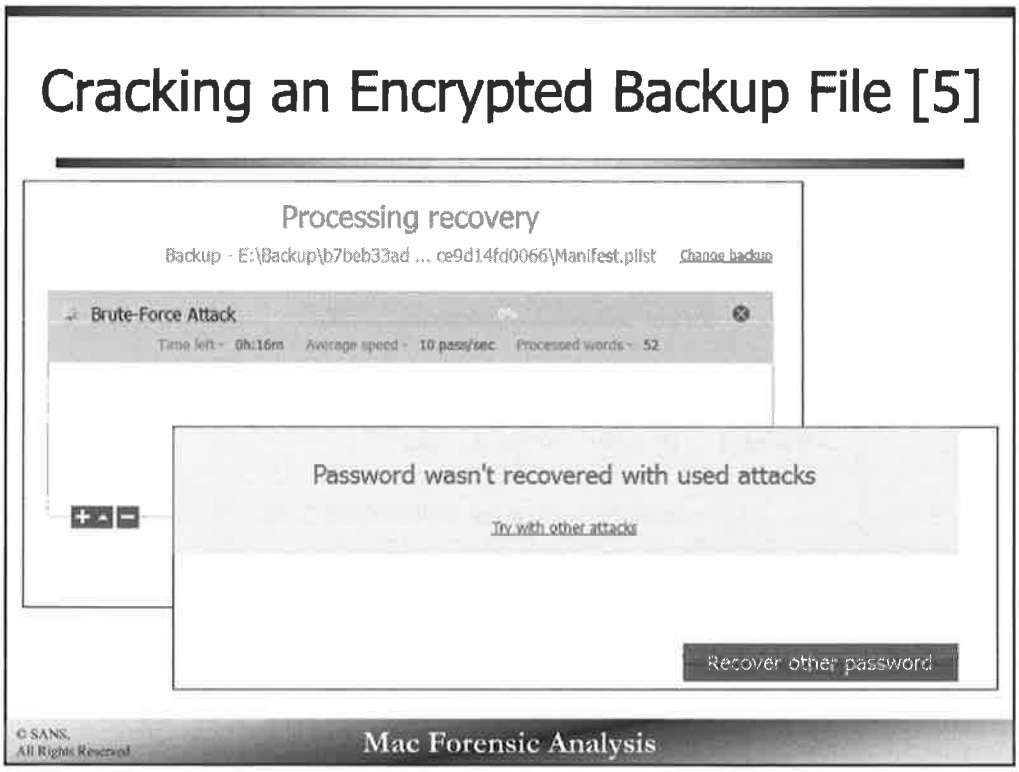
If a manifest.plist file is selected that isn't encrypted, an error message will appear. When forensic tools cannot access a backup file it is commonly associated with the backup being encrypted. Should the tools fail at parsing and EPPB report that the backup is not encrypted, it is likely that the backup file is corrupt. EPPB will only crack encrypted backup files and cannot repair corrupt backups.



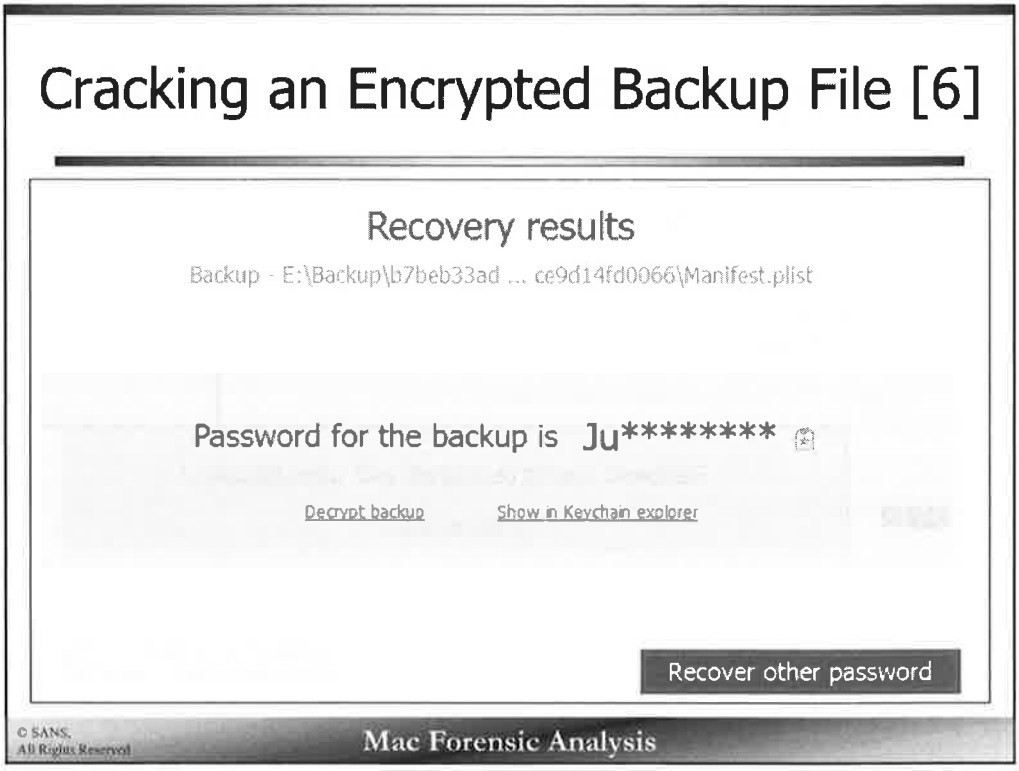


Once the encrypted backup file is loaded in the EPPB, the examiner can select to run dictionary attacks (where dictionary files can be loaded) or to conduct a brute force attack. Both methods allow for customized settings by the examiner as shown below.

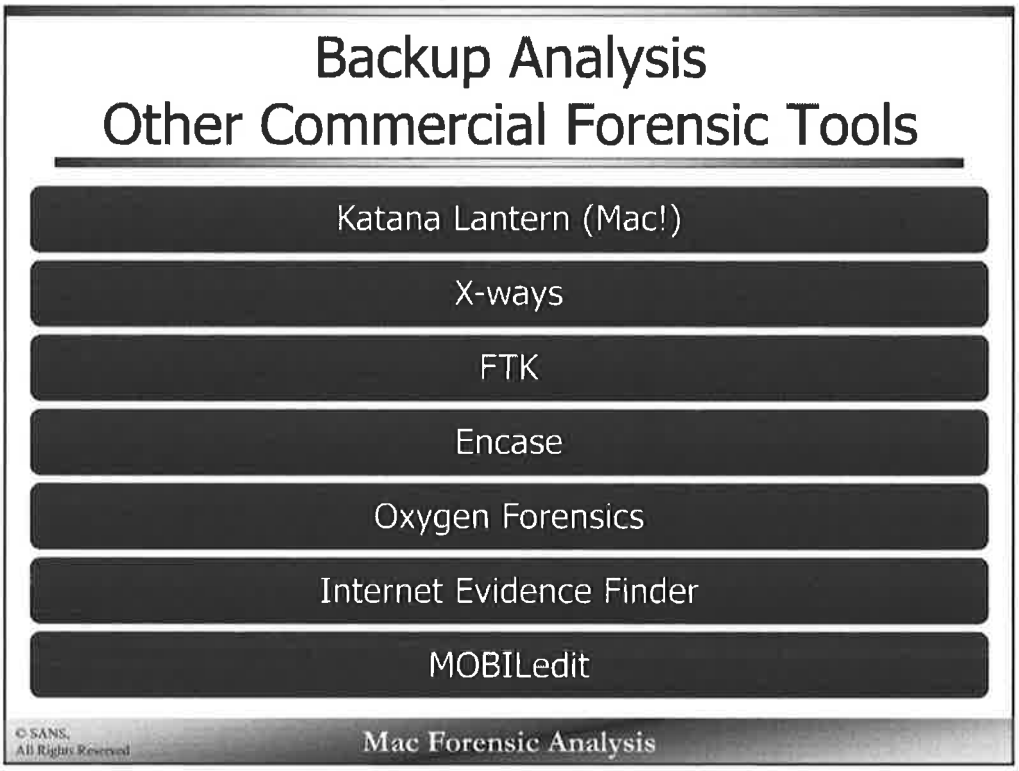




When an attack is selected and fails, an error message appears alerting the examiner to select other attack methods. A brute force attack is the fastest when 1-4 characters ranging between [0-9] is selected.



The example of the password cracking shown below was successful with a dictionary file pulled from another device of the same user. Most users use common passwords amongst their devices, so it's wise to use any dictionary or keyboard cache files to attempt dictionary attacks in EPPB.

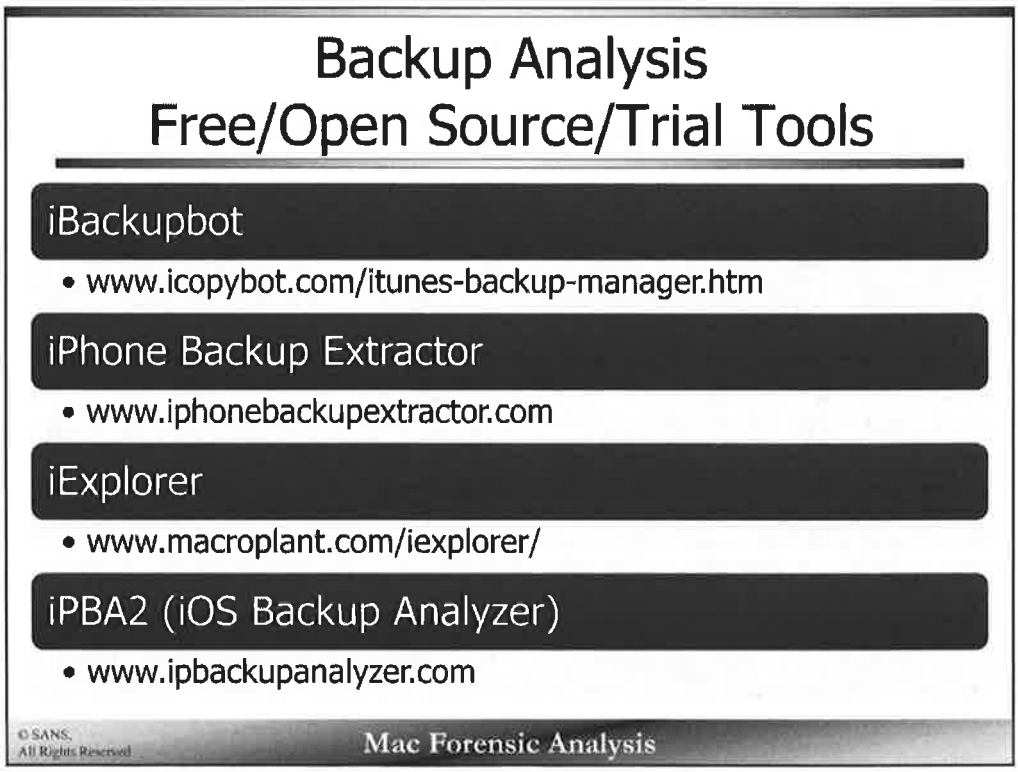


Many commercial mobile acquisition tools exist to analyze backup files.

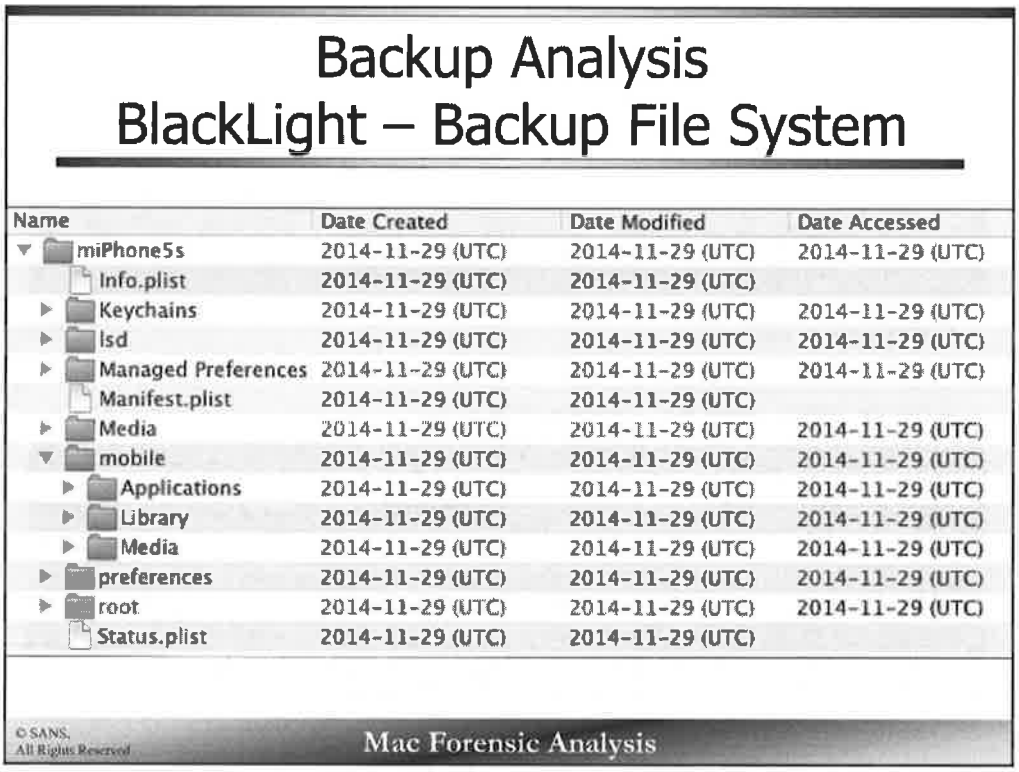
References:

http://www.oxygen-forensic.com/en/features/analyst/backup-reader/itunes-backup

http://www.magnetforensics.com/investigating-ios-phone-images-file-dumps-backups/



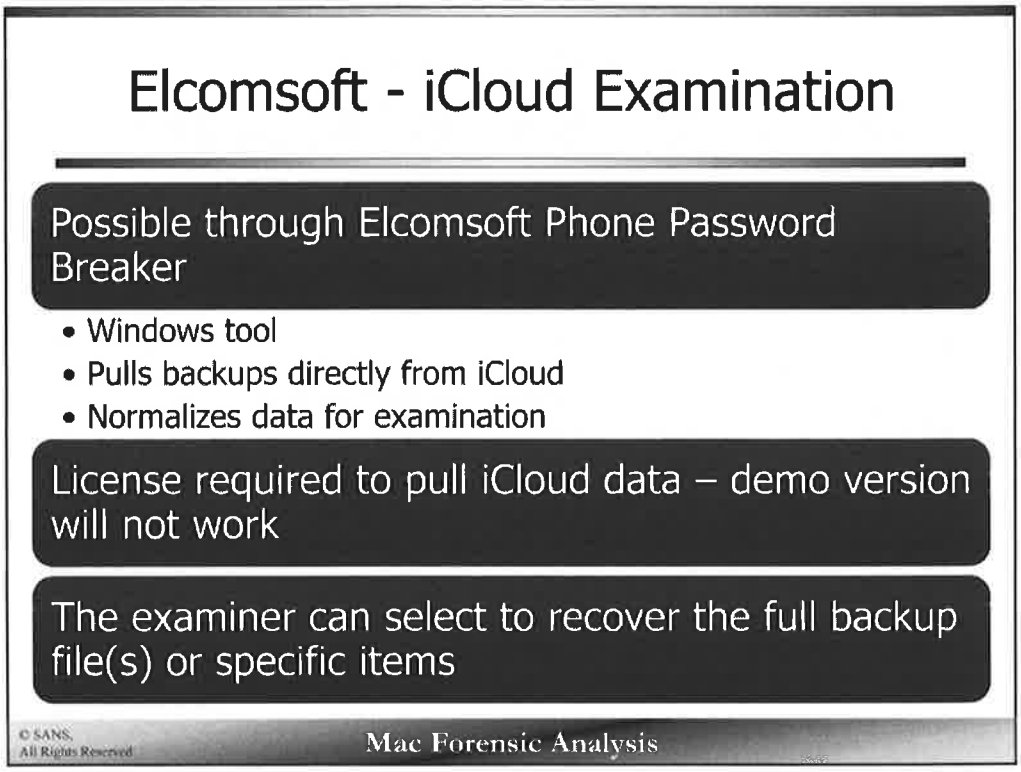
Many free/open source/trial mobile acquisition tools exist to analyze backup files.



Once opened, a backup file will look very similar to a File System acquisition of a device.

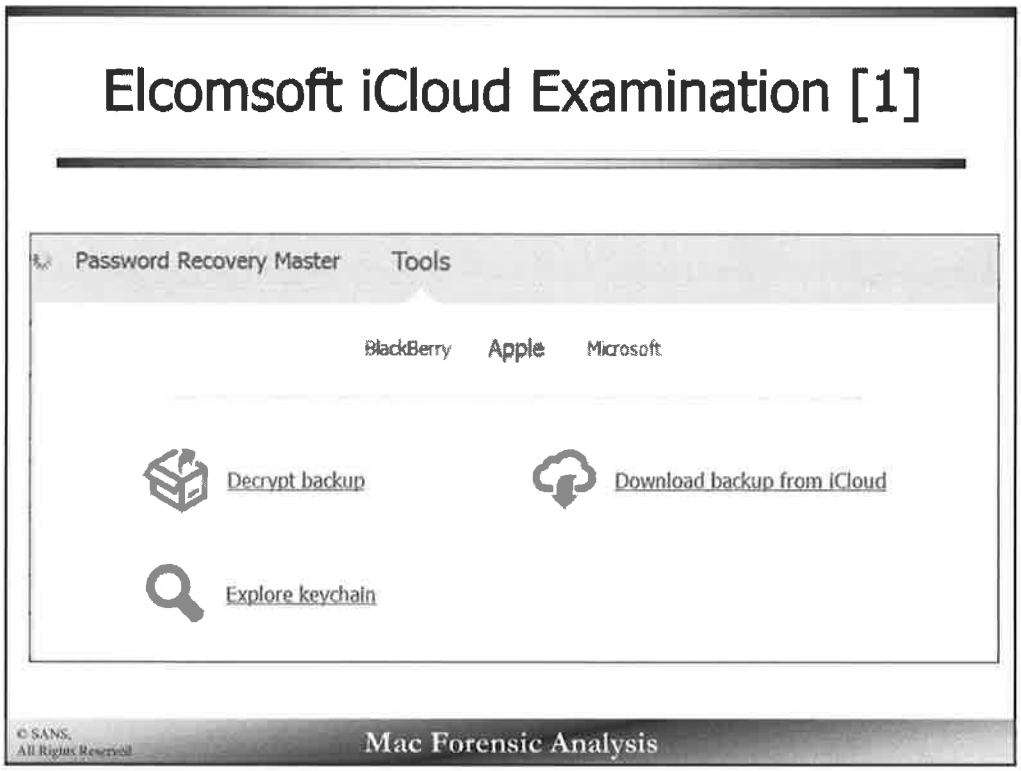
The mobile directory will by far have the most user created data in its three main directories:

- Applications
- Library
- Media

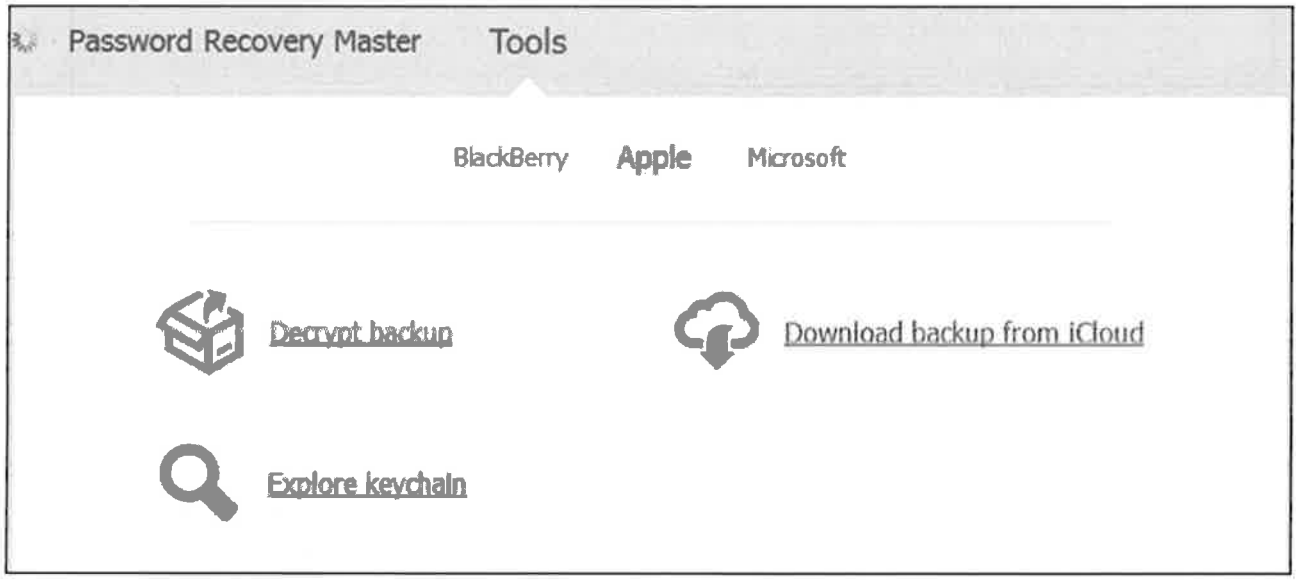


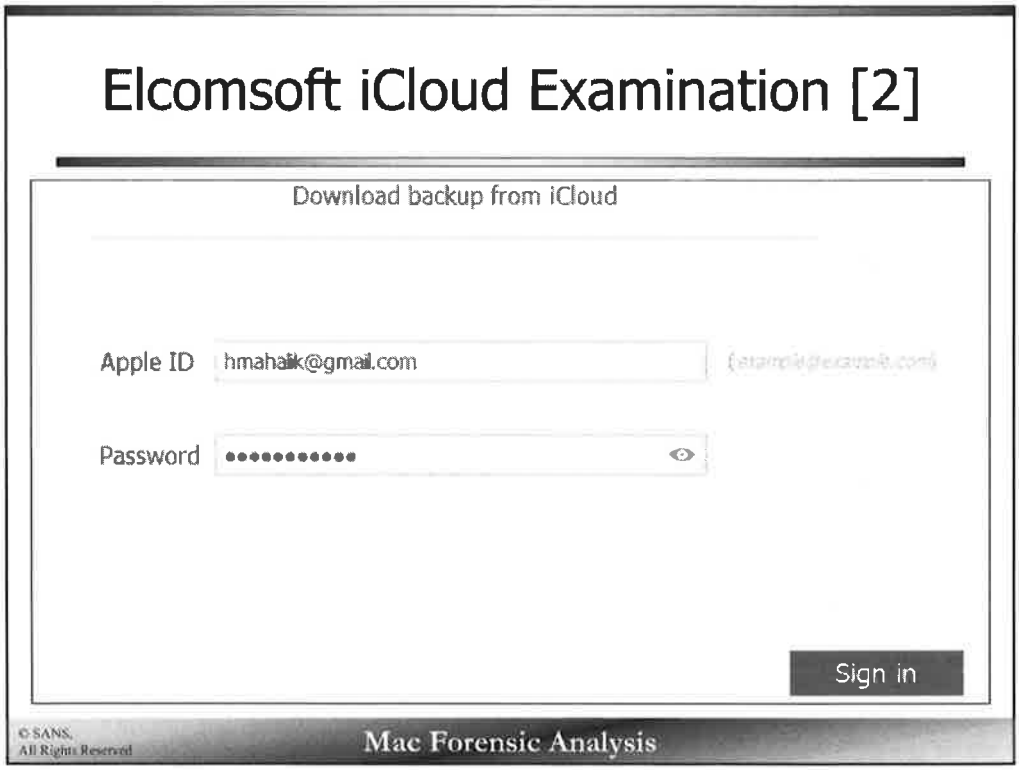
iCloud data can be parsed and decrypted using Elcomsoft Phone Password Breaker (EPPB). EPPB pulls the backups from iCloud directly when the examiner enters the Apple ID and password to access the backups. A full license is required to access iCloud data. The examiner can select to pull all contents of the backup(s) or to manually select the items for which they want to recover. Once the backup files is downloaded, the backup can be decrypted using EPPB. This allows the user to view a normalized folder structure for ease of examination.

http://www.elcomsoft.com/eppb.html

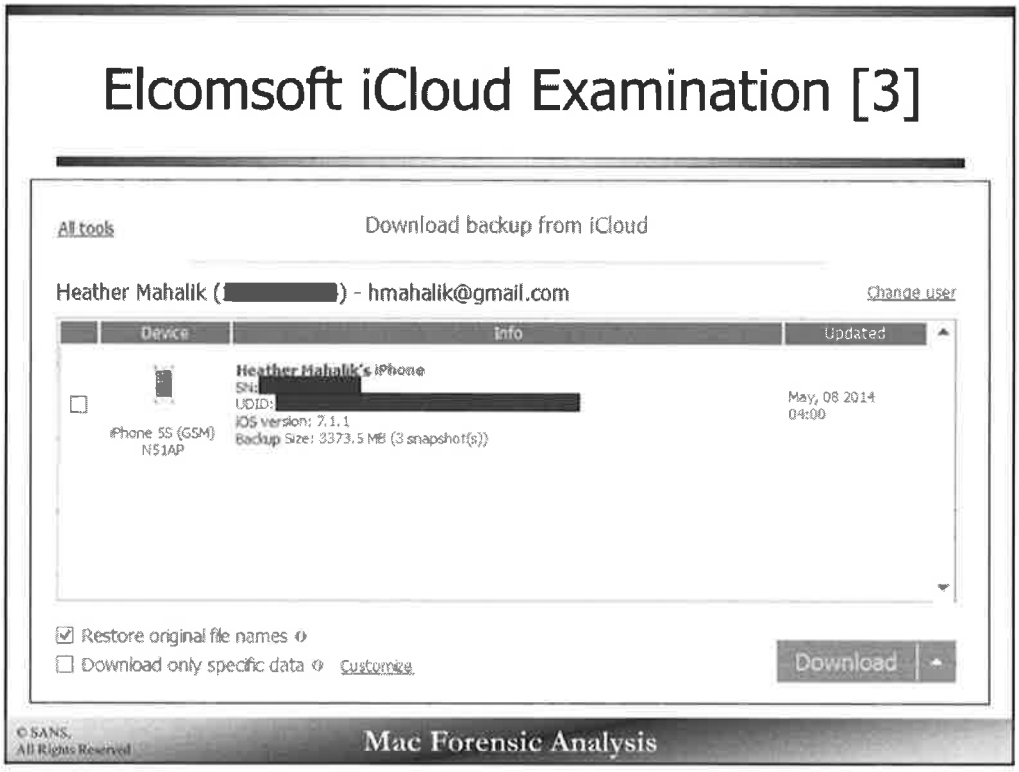


Under the Tools option in EPPB, the examiner must select to "Download backup from iCloud."

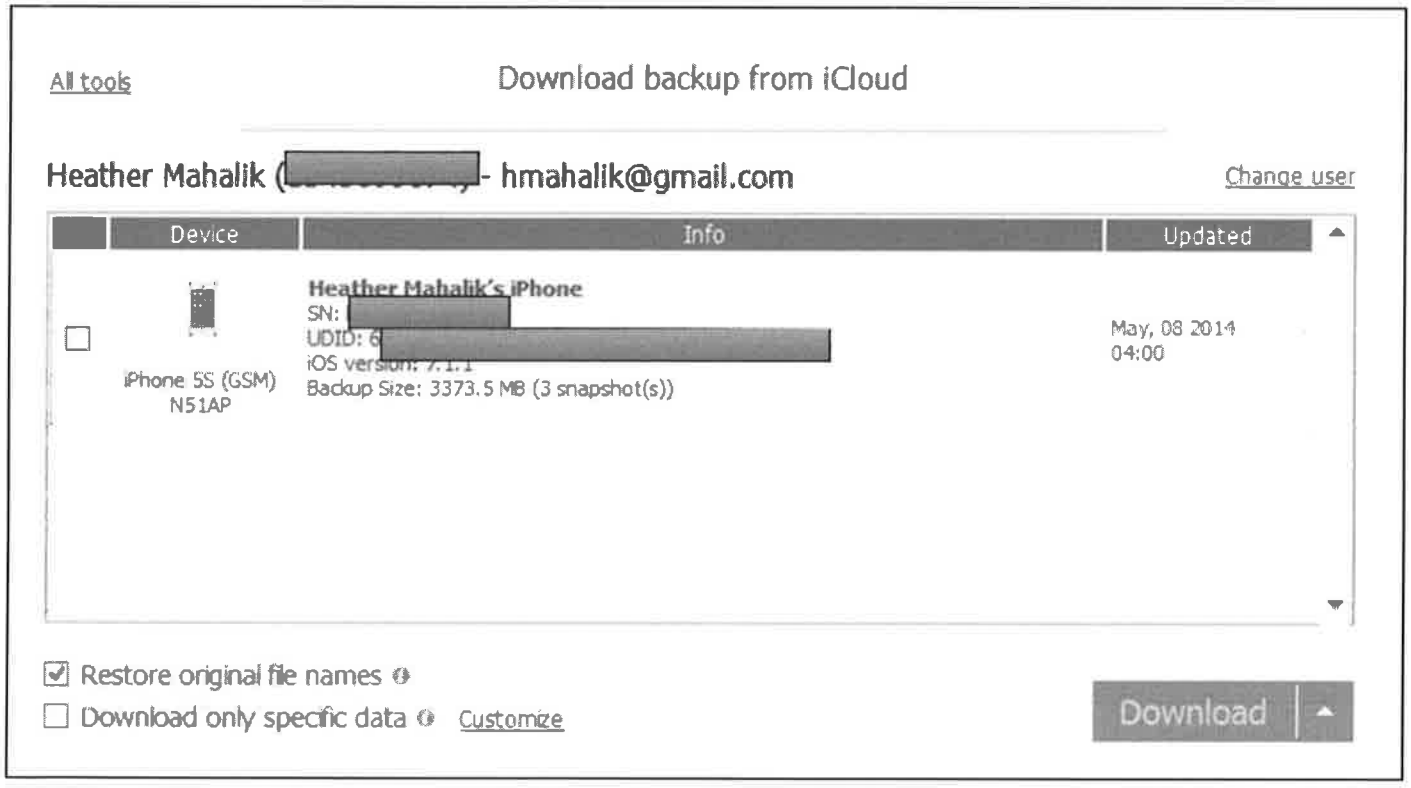


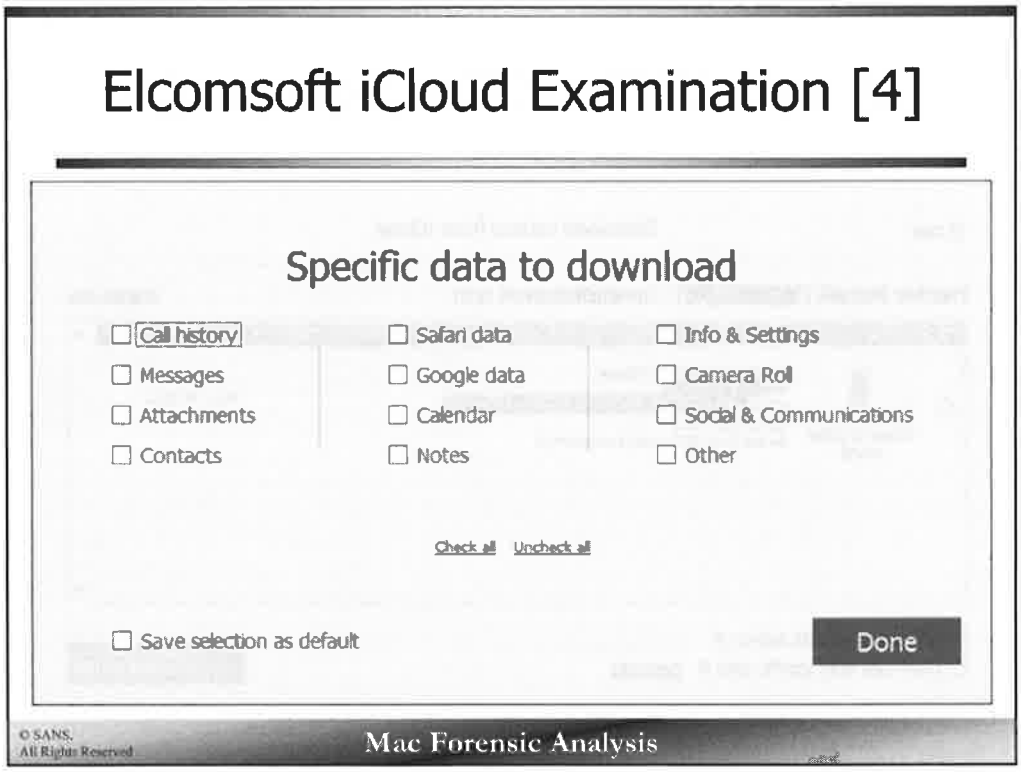


EPPB prompts the examiner to enter the Apple ID and password associated with the iCloud account.

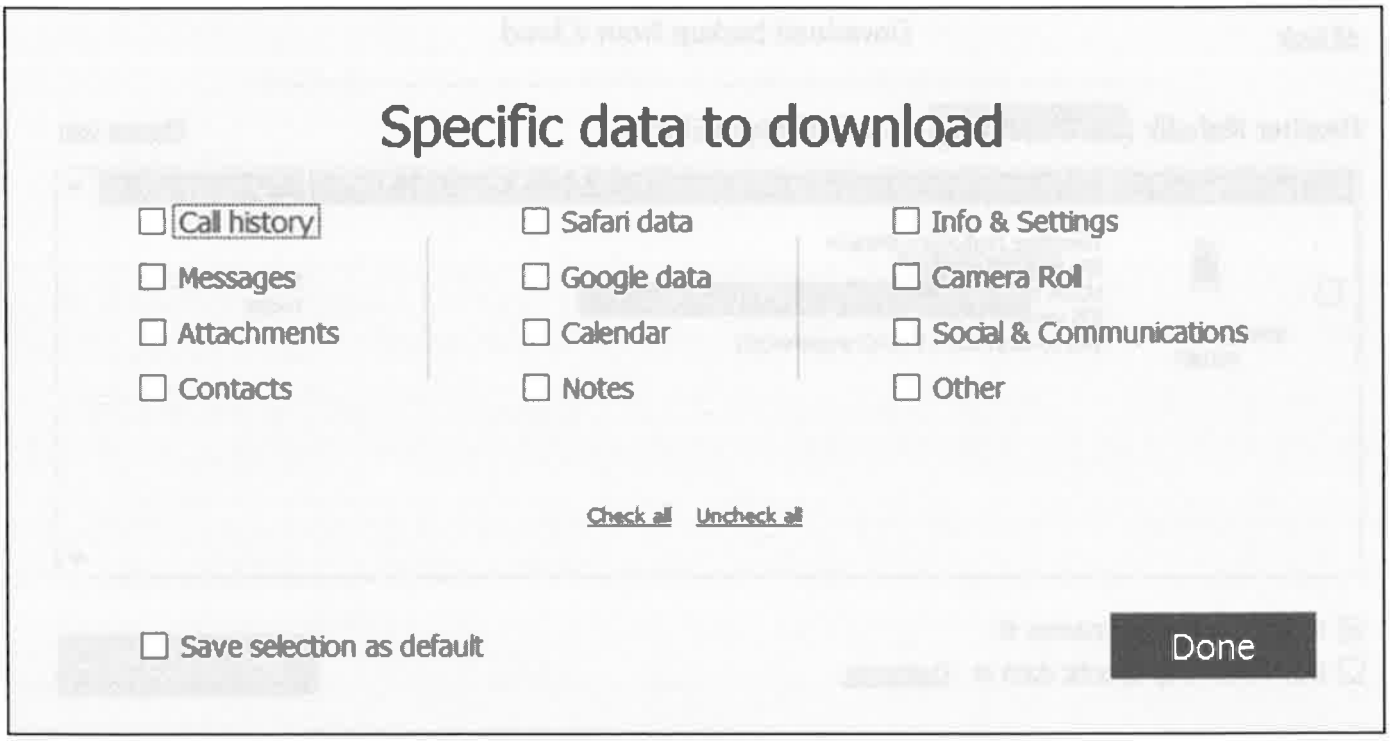


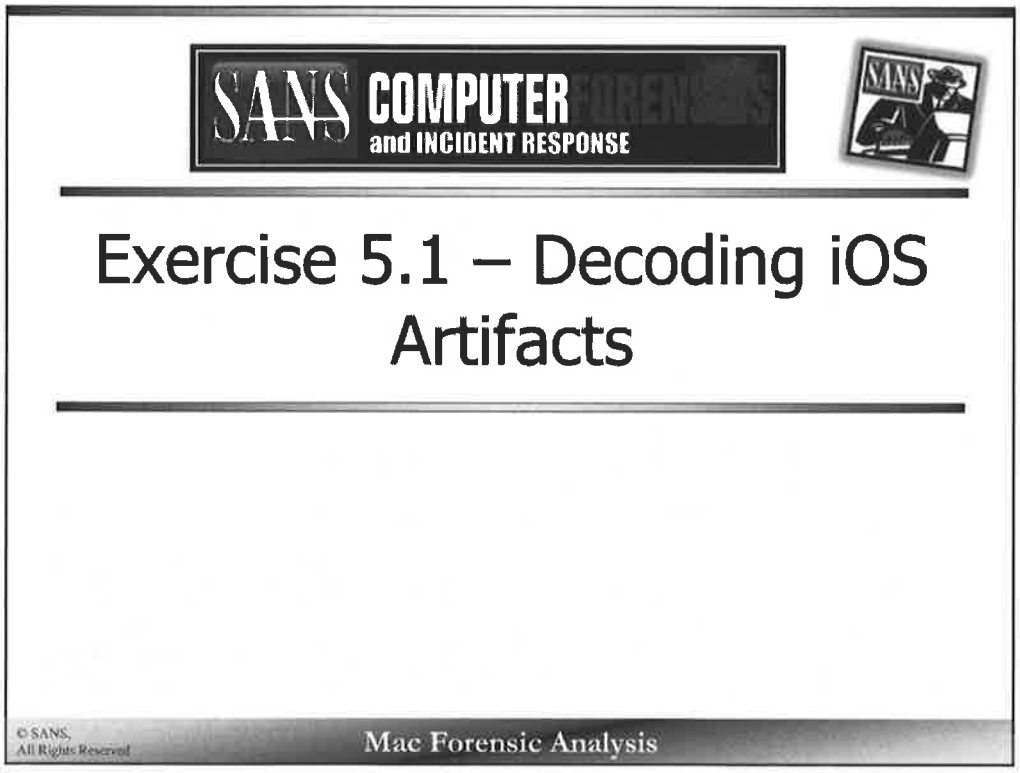
All iOS devices associate with the Apple ID and password will be listed. The examiner can select to pull backups from multiple devices and can customize the data selected from each as shown in the next slide.



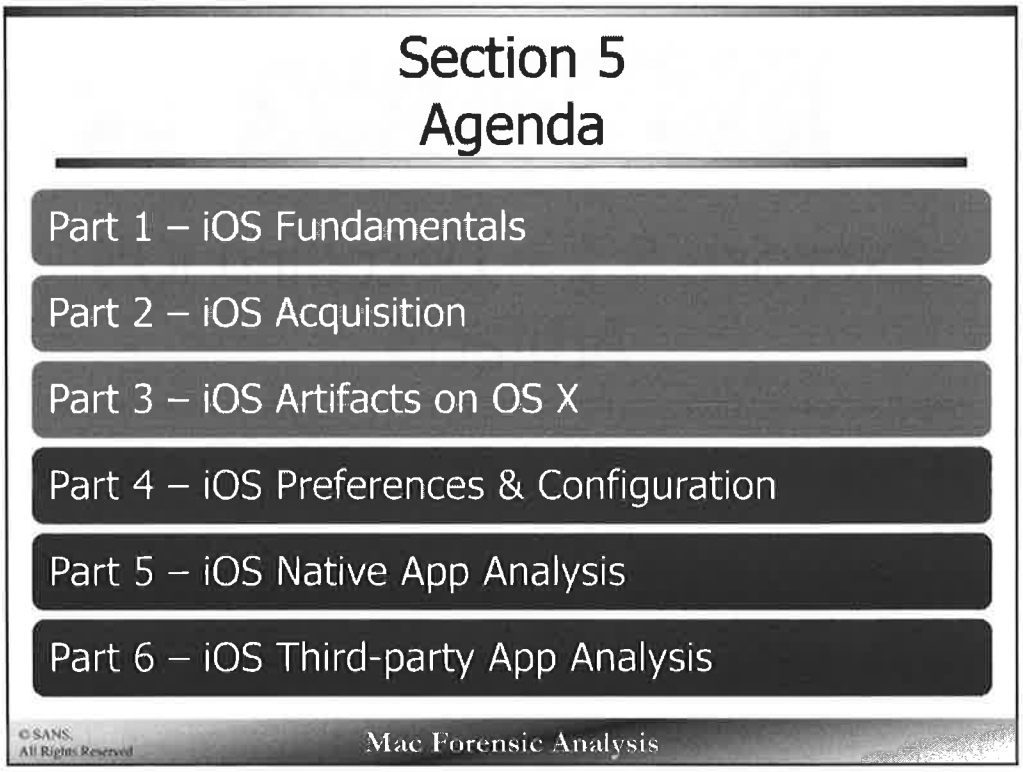


When the examiner elects to customize the data pulled from the iCloud backup, the following options are provided. Note: Other will pull any data not listed that is accessible.

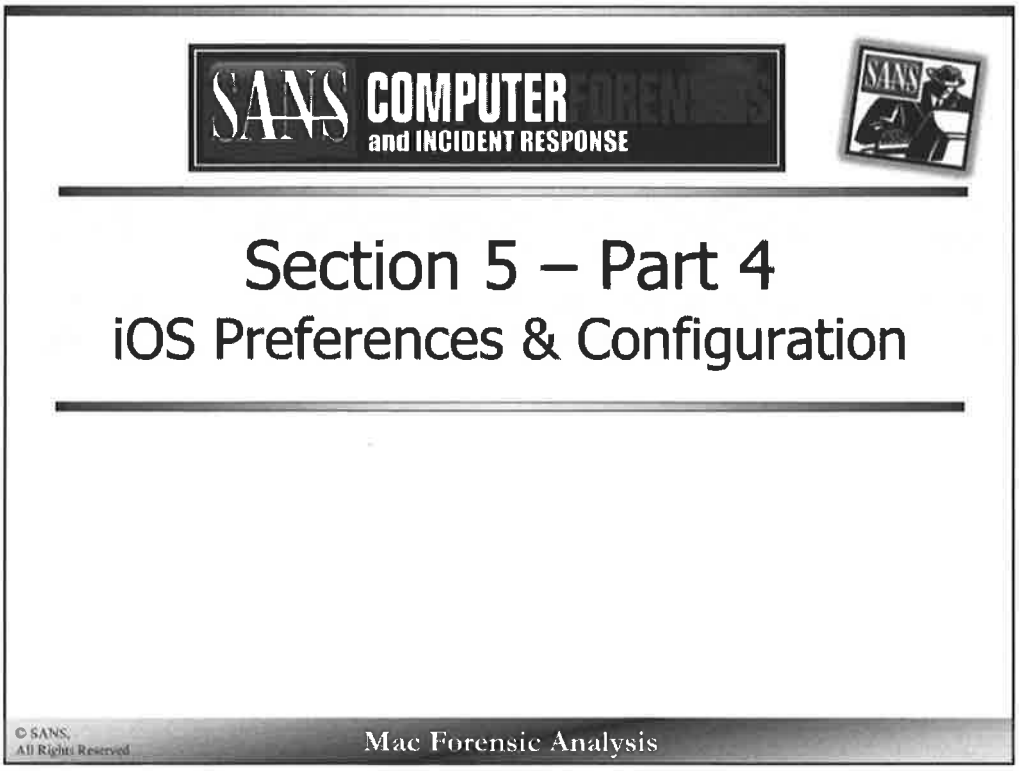




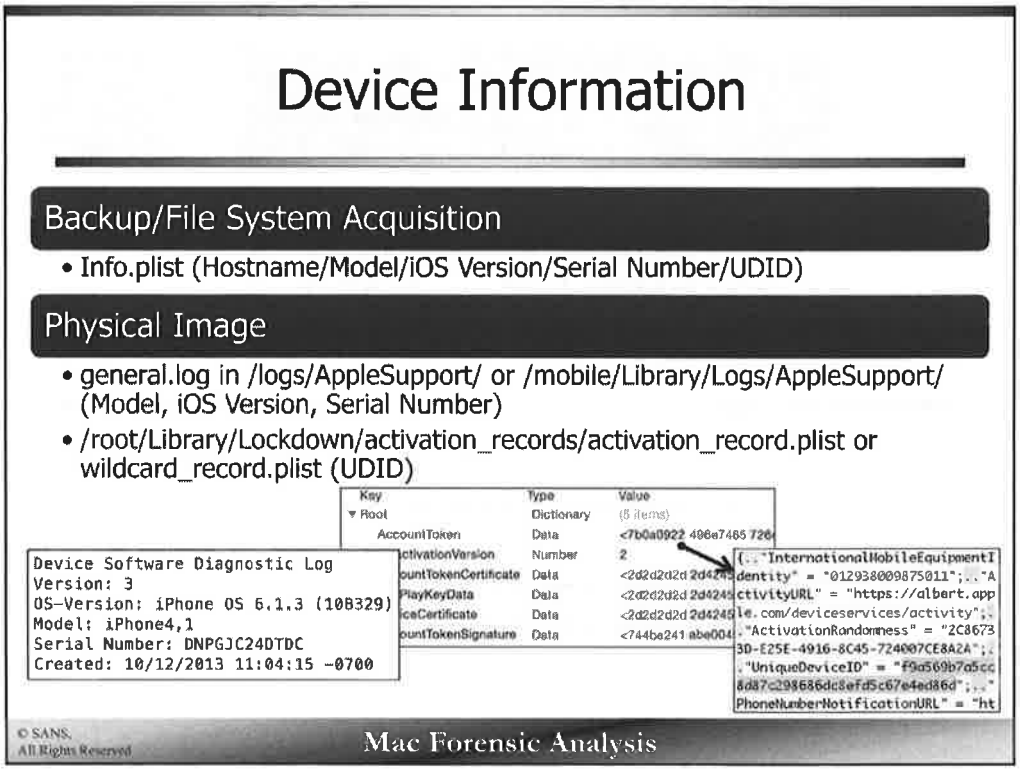
This page intentionally left blank.



This page intentionally left blank.



This page intentionally left blank.



Simple device information such as model, iOS version, device serial number, UDID, and hostname can be found in a variety of locations depending if you have a file system/backup acquisition or a full physical image.

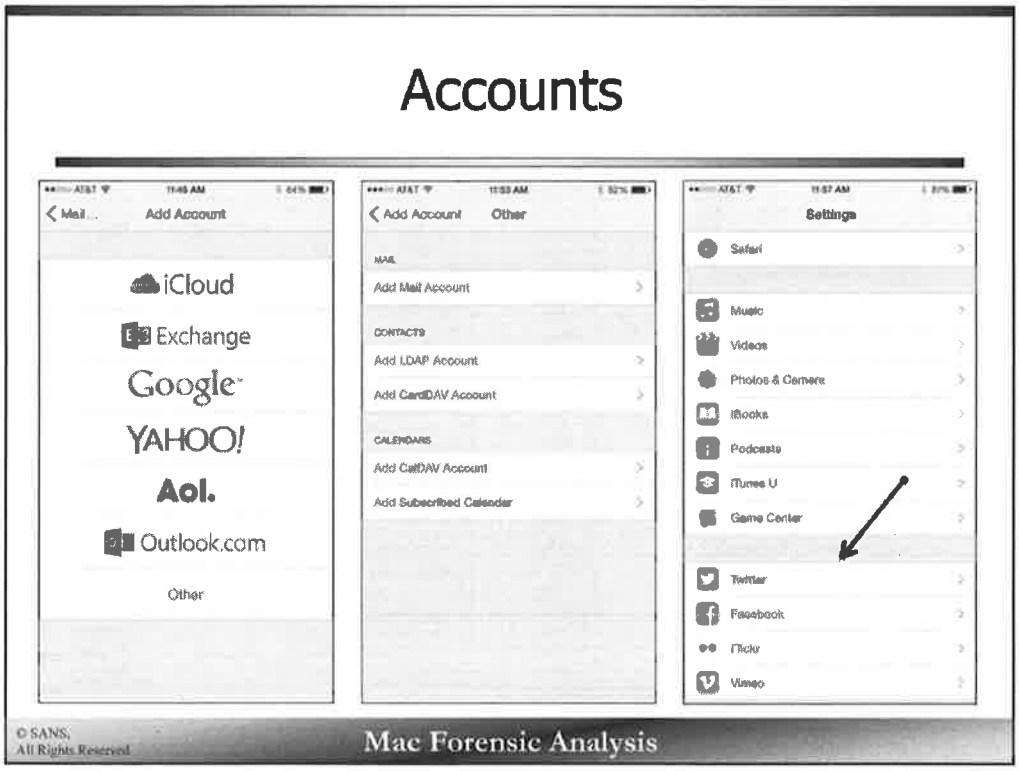
On backups or file system acquisitions, this information can easily be found in the `Info.plist` that is created. Some acquisition tools do not necessarily call this the `Info.plist` but should have a similar files containing the identifying information for the device.

On physical images, this information is stored in many files:

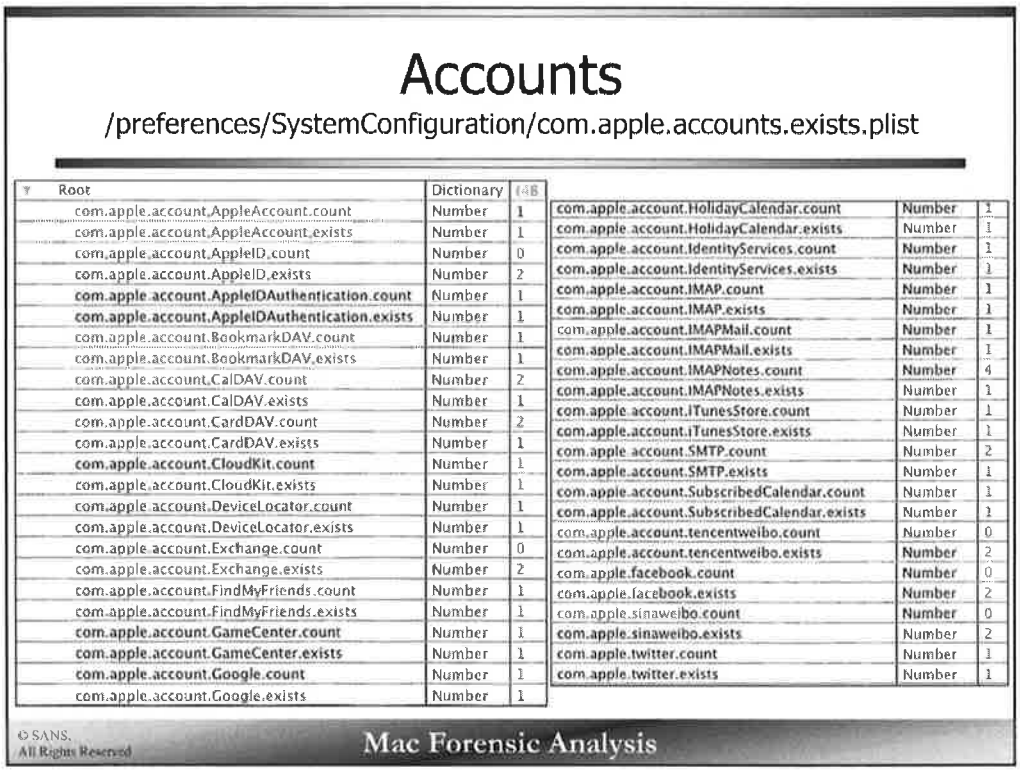
- Model, iOS Version, Serial Number can be found in the `general.log` file located in `/logs/AppleSupport/` or `/mobile/Library/Logs/AppleSupport/` as the header of the log file.
- The UDID can be found in the `activation_record.plist` or `wildcard_record.plist` file located in `/root/Library/Lockdown/` directory in the as embedded data in `AccountToken` key.

Other files on the device contain similar information:

- Model: `/preferences/SystemConfiguration/NetworkInterfaces.plist`
- Model, Hostname: `/preferences/SystemConfiguration/preferences.plist`
- Hostname: `/preferences/SystemConfiguration/com.apple.mobilegestalt.plist`



Knowing what accounts a user has configured with their device can be good investigative information to know. The `com.apple.accounts.exists.plist` file located in the `/preferences/SystemConfiguration/` directory for backups, file system extractions, and physical images.



Knowing what accounts a user has configured with their device can be good investigative information to know. The `com.apple.accounts.exists.plist` file located in the `/preferences/SystemConfiguration/` directory for backups, file system extractions, and physical images.

This property list shows which type of accounts are globally configured on the device, everything from e-mail (IMAP/SMTP/Exchange), Google, Calendar, Facebook, Twitter, Apple, Sinaweibo, etc. Each account has two associated keys, a "Count" and an "Exists" key.

The "`Exists`" key shows if that particular type of account is in use. This can have a few different options:

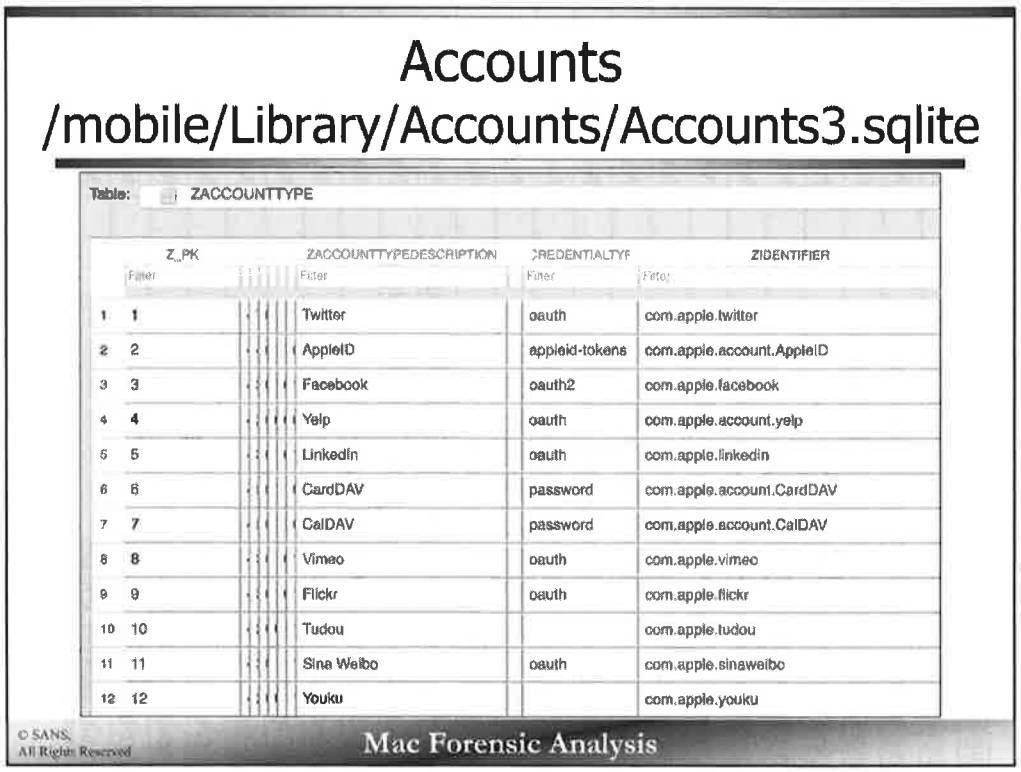
- 1 – At least one account is setup for this type
- 2 – No accounts of this type

The "`Count`" key shows how many accounts of this type there are. For example, in the screenshot above this device has:

- Two SMTP e-mail accounts
- One Apple Account, Google, Twitter, IMAP
- No Exchange, Facebook, Sinaweibo, Tencentweibo accounts

The example above shows an device running iOS 8, older iOS versions had fewer accounts listed – mainly AppleID, Facebook, and Sinoweibo.

Root	Dictionary	148
com.apple.account.AppleAccount.count	Number	1
com.apple.account.AppleAccount.exists	Number	1
com.apple.account.AppleID.count	Number	0
com.apple.account.AppleID.exists	Number	2
com.apple.account.AppleIDAuthentication.count	Number	1
com.apple.account.AppleIDAuthentication.exists	Number	1
com.apple.account.BookmarkDAV.count	Number	1
com.apple.account.BookmarkDAV.exists	Number	1
com.apple.account.CalDAV.count	Number	2
com.apple.account.CalDAV.exists	Number	1
com.apple.account.CardDAV.count	Number	2
com.apple.account.CardDAV.exists	Number	1
com.apple.account.CloudKit.count	Number	1
com.apple.account.CloudKit.exists	Number	1
com.apple.account.DeviceLocator.count	Number	1
com.apple.account.DeviceLocator.exists	Number	1
com.apple.account.Exchange.count	Number	0
com.apple.account.Exchange.exists	Number	2
com.apple.account.FindMyFriends.count	Number	1
com.apple.account.FindMyFriends.exists	Number	1
com.apple.account.GameCenter.count	Number	1
com.apple.account.GameCenter.exists	Number	1
com.apple.account.Google.count	Number	1
com.apple.account.Google.exists	Number	1
com.apple.account.HolidayCalendar.count	Number	1
com.apple.account.HolidayCalendar.exists	Number	1
com.apple.account.IdentityServices.count	Number	1
com.apple.account.IdentityServices.exists	Number	1
com.apple.account.IMAP.count	Number	1
com.apple.account.IMAP.exists	Number	1
com.apple.account.IMAPMail.count	Number	1
com.apple.account.IMAPMail.exists	Number	1
com.apple.account.IMAPNotes.count	Number	4
com.apple.account.IMAPNotes.exists	Number	1
com.apple.account.iTunesStore.count	Number	1
com.apple.account.iTunesStore.exists	Number	1
com.apple.account.SMTP.count	Number	2
com.apple.account.SMTP.exists	Number	1
com.apple.account.SubscribedCalendar.count	Number	1
com.apple.account.SubscribedCalendar.exists	Number	1
com.apple.account.tencentweibo.count	Number	0
com.apple.account.tencentweibo.exists	Number	2
com.apple.facebook.count	Number	0
com.apple.facebook.exists	Number	2
com.apple.sinaweibo.count	Number	0
com.apple.sinaweibo.exists	Number	2
com.apple.twitter.count	Number	1
com.apple.twitter.exists	Number	1

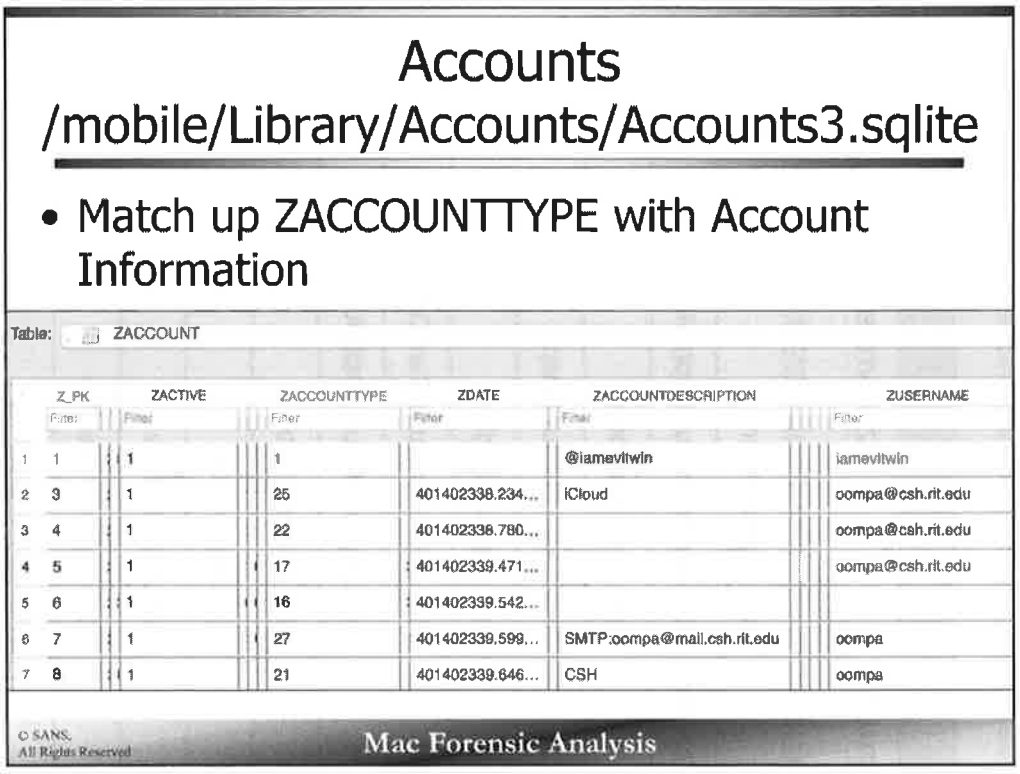


The `Accounts3.sqlite` database in `/mobile/Library/Accounts/` contains more specific account information.

The `ZACCOUNTYPE` table contains the types of accounts that can be configured globally on the device. The `Z_PK` column contains the identification number for each account that we will be looking at in upcoming slides. This table also includes the account description, credential types, and the identifier for the account.

Table: ZACCOUNTTYPE

	Z_PK	ZACCOUNTTYPEDESCRIPTION	CREDENTIALTYF	ZIDENTIFIER
	Filter	Filter	Filter	Filter
1	1	Twitter	oauth	com.apple.twitter
2	2	AppleID	appleid-tokens	com.apple.account.AppleID
3	3	Facebook	oauth2	com.apple.facebook
4	4	Yelp	oauth	com.apple.account.yelp
5	5	LinkedIn	oauth	com.apple.linkedin
6	6	CardDAV	password	com.apple.account.CardDAV
7	7	CalDAV	password	com.apple.account.CalDAV
8	8	Vimeo	oauth	com.apple.vimeo
9	9	Flickr	oauth	com.apple.flickr
10	10	Tudou		com.apple.tudou
11	11	Sina Weibo	oauth	com.apple.sinaweibo
12	12	Youku		com.apple.youku



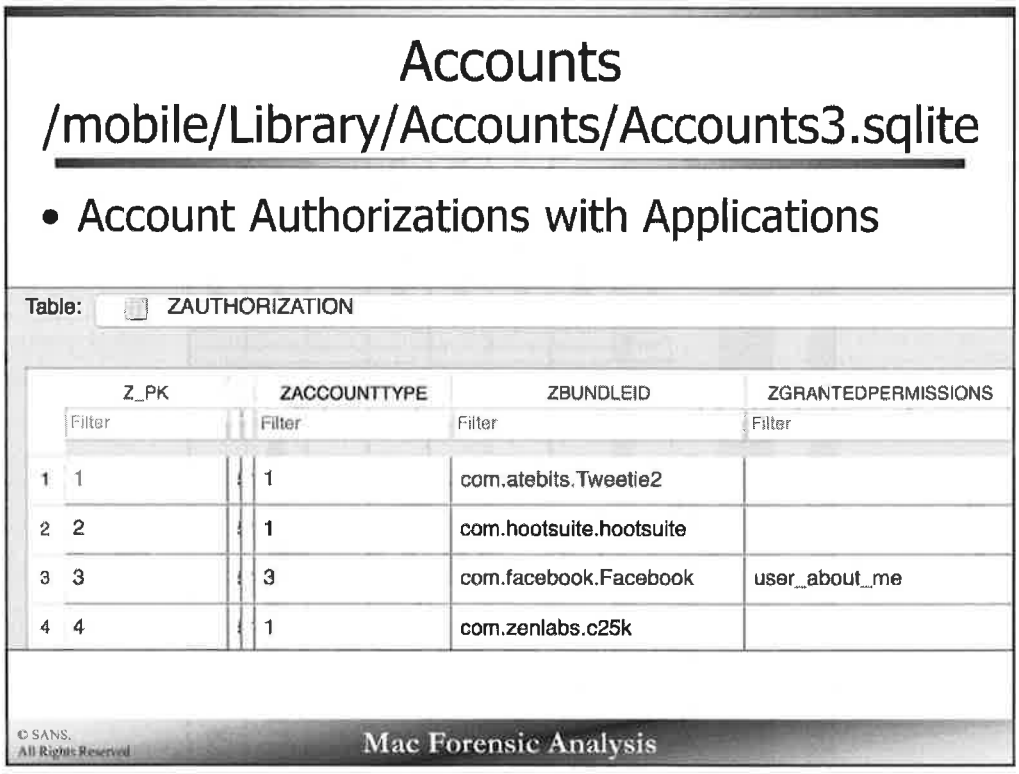
The ZACCOUNT table contains specific user account information. We can match up the Z_PK key from the ZACCOUNTTYPES table with the ZACCOUNTTYPE column of this table to determine what account this information is associated with.

For example the first tuple, is an account type of "1", looking at the previous slide we can see that this is a Twitter account. The user account information and description used is in this table. We can see this user has many accounts:

- 1 – Twitter (@iamevltwin)
- 25 – iCloud (oompa@csh.rit.ed)
- 22 – Messages (oompa@csh.rit.edu)
- 17 – Device Locator (oompa@csh.rit.edu
- 16 – IMAPMail
- 27 – SMTP (oompa)
- 21 – IMAP (oompa)

Table: ZACCOUNT

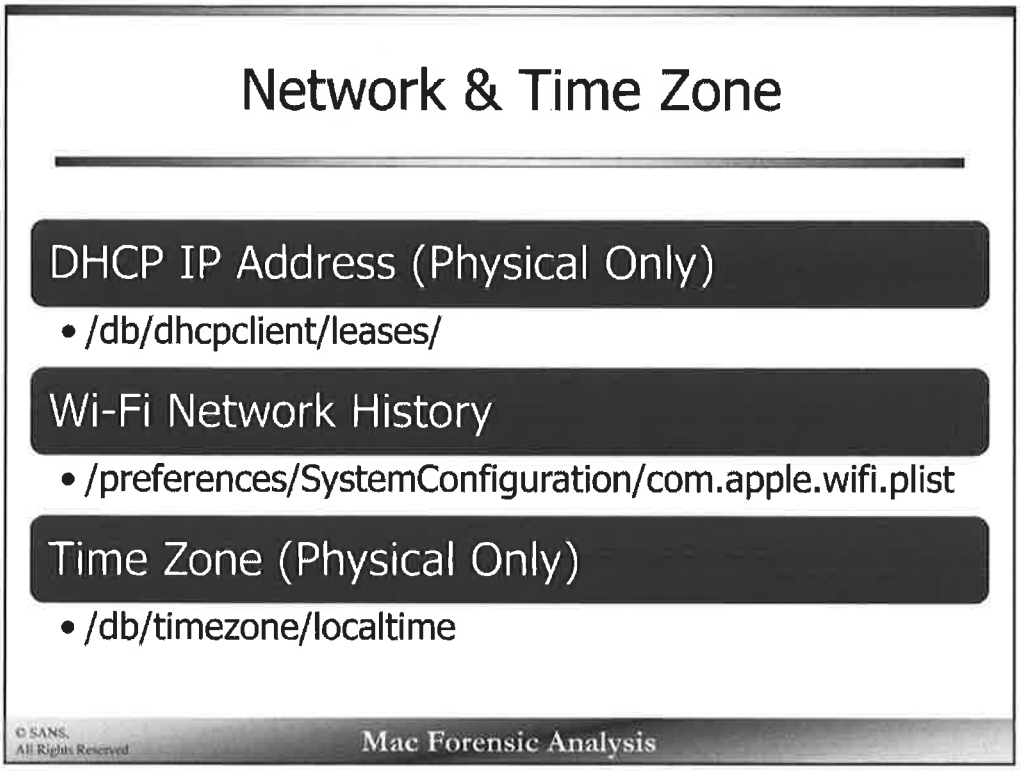
	Z_PK	ZACTIVE	ZACCOUNTTYPE	ZDATE	ZACCOUNTDESCRIPTION	ZUSERNAME
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	1	1		@iamevitwin	iamevitwin
2	3	1	25	401402338.234...	iCloud	oompa@csh.rit.edu
3	4	1	22	401402338.780...		oompa@csh.rit.edu
4	5	1	17	401402339.471...		oompa@csh.rit.edu
5	6	1	16	401402339.542...		
6	7	1	27	401402339.599...	SMTP:oompa@mail.csh.rit.edu	oompa
7	8	1	21	401402339.646...	CSH	oompa



The `ZAUTHORIZATION` table contains information associated if a specific account has access to an application.

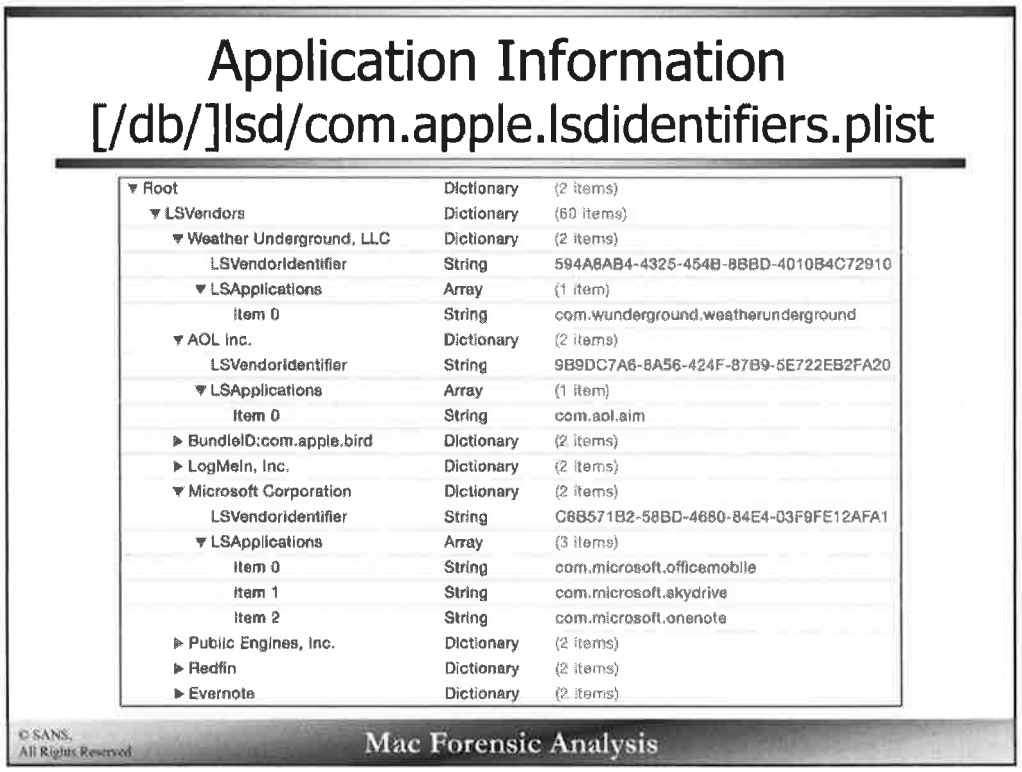
In the example above, the Twitter account has authorization to use the three applications:

- `com.atebits.Tweetie2` – Twitter App
- `com.hootsuite.hootsuite` – Hootsuite App
- `com.zenlabs.c25k` – Couch to 5k Application



Very similar to OS X systems. The `com.apple.wifi.plist` is similar in format to the `com.apple.airport.preferences.plist` on OS X.

If the time zone was never explicitly set to a specific time zone using the interface, the `/db/timezone/localtime` file may not be created.



A quick way to determine which apps are installed on a device is to use the `com.apple.lsidentifiers.plist` file. This file contains a friendly company name, an app identifier, and GUID app identifiers.

This property list will be in a slightly different location depending on the type of acquisition:

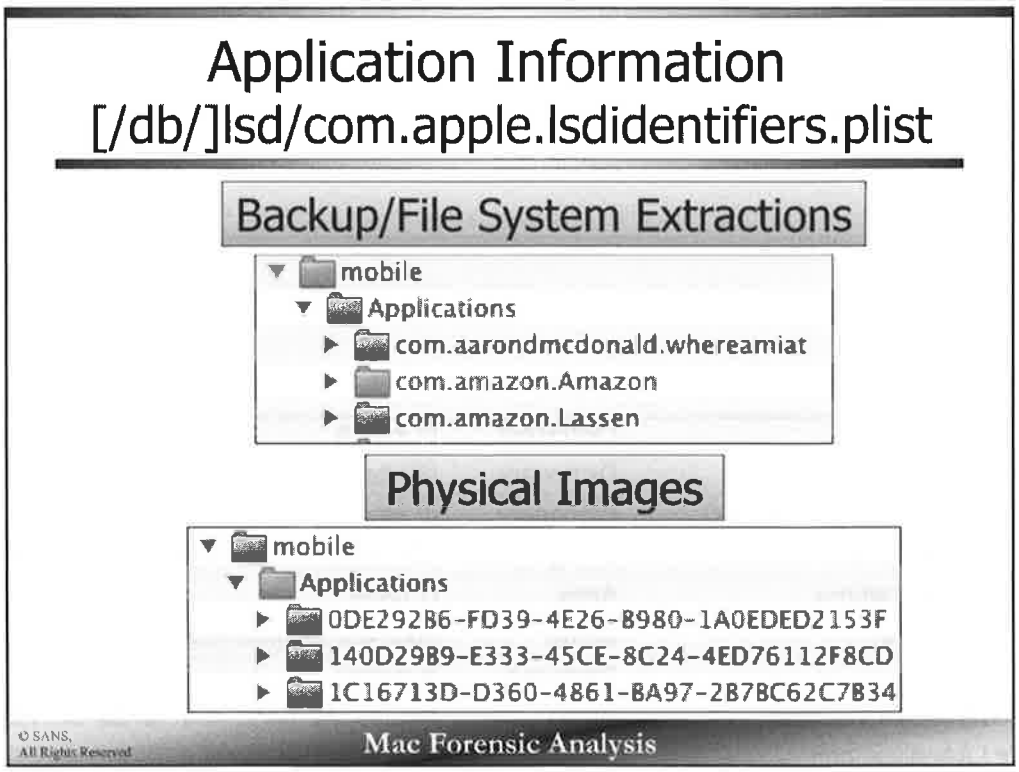
- Physical - `/db/lsd/com.apple.lsdidentifiers.plist`
- Backup/File System - `/lsd/com.apple.lsdidentifiers.plist`

The property list is organized by vendor, with at least one application's information underneath. Each application has an associated bundle ID (reverse DNS notation) and an associated GUID identifier. This identifier is the directory that is used on the live device (or physical images) to organize application data for that specific application.

If a device has multiple applications from the same vendor, you will see more than one applications and its associated information under each Vendor's name. An example of this in the screen shot is Microsoft where the device has three apps installed by this vendor:

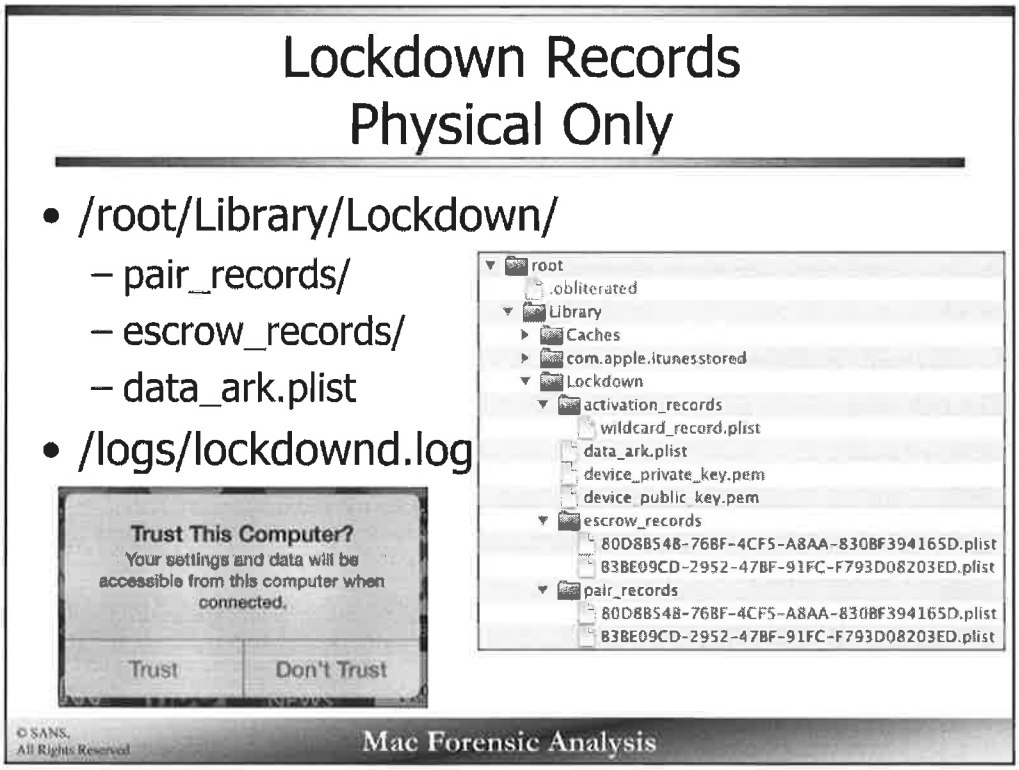
- Office Mobile
- OneDrive (previously SkyDrive)
- OneNote

▼ Root	Dictionary	(2 items)
▼ LSVendors	Dictionary	(60 items)
▼ Weather Underground, LLC	Dictionary	(2 items)
LSVendorIdentifier	String	594A8AB4-4325-454B-8BBD-4010B4C72910
▼ LSApplications	Array	(1 item)
Item 0	String	com.wunderground.weatherunderground
▼ AOL Inc.	Dictionary	(2 items)
LSVendorIdentifier	String	9B9DC7A6-6A56-424F-87B9-5E722EB2FA20
▼ LSApplications	Array	(1 item)
Item 0	String	com.aol.aim
▶ BundleID:com.apple.bird	Dictionary	(2 items)
▶ LogMeIn, Inc.	Dictionary	(2 items)
▼ Microsoft Corporation	Dictionary	(2 items)
LSVendorIdentifier	String	C8B571B2-58BD-4680-84E4-03F9FE12AFA1
▼ LSApplications	Array	(3 items)
Item 0	String	com.microsoft.officemobile
Item 1	String	com.microsoft.skydrive
Item 2	String	com.microsoft.onenote
▶ Public Engines, Inc.	Dictionary	(2 items)
▶ Redfin	Dictionary	(2 items)
▶ Evernote	Dictionary	(2 items)



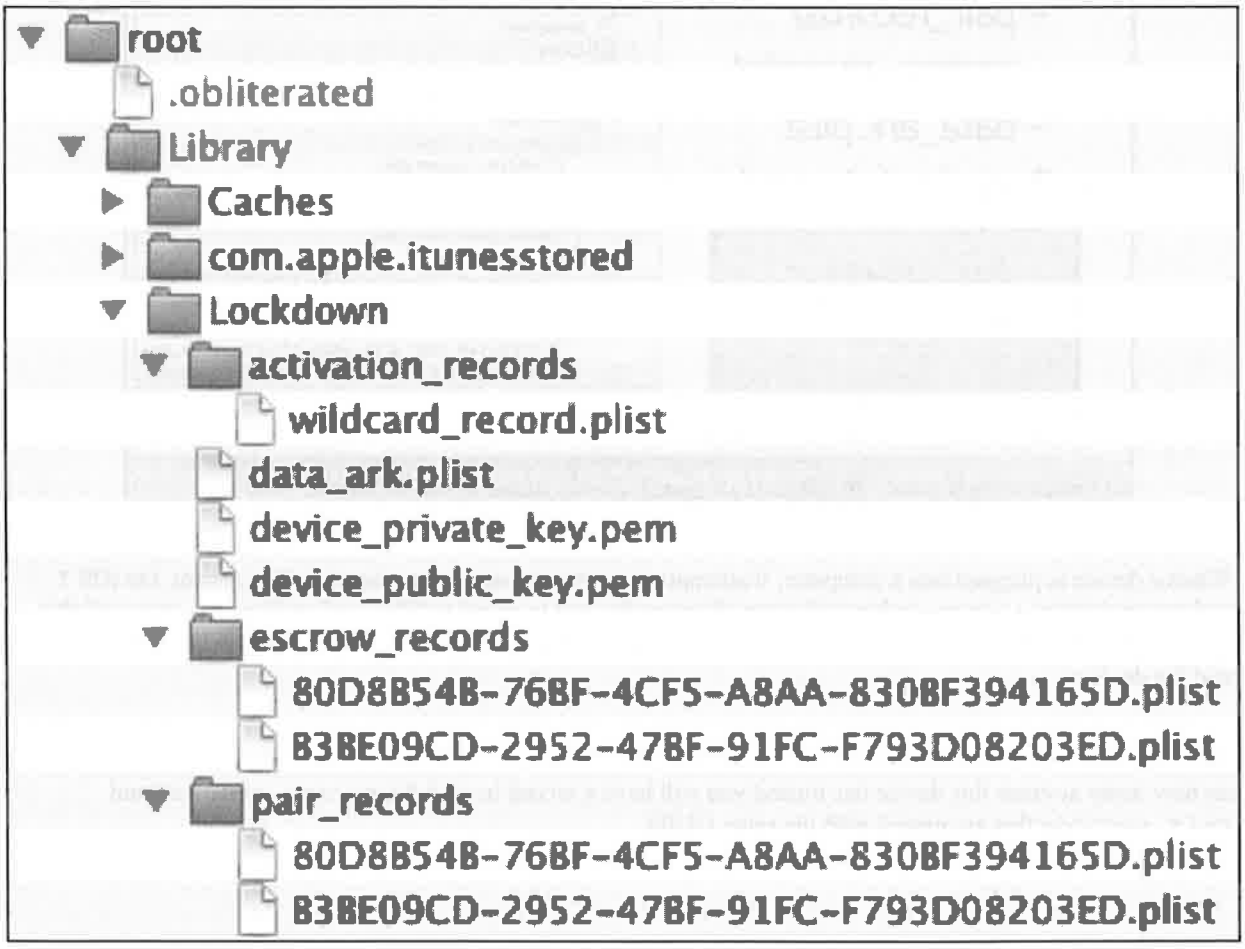
Backup and File System extractions use the information in the `com.apple.lsdidentifiers.plist` to normalize the applications directory shown to the forensic analyst. In these you will see the bundle identifiers rather than the no-so-human friendly GUID.

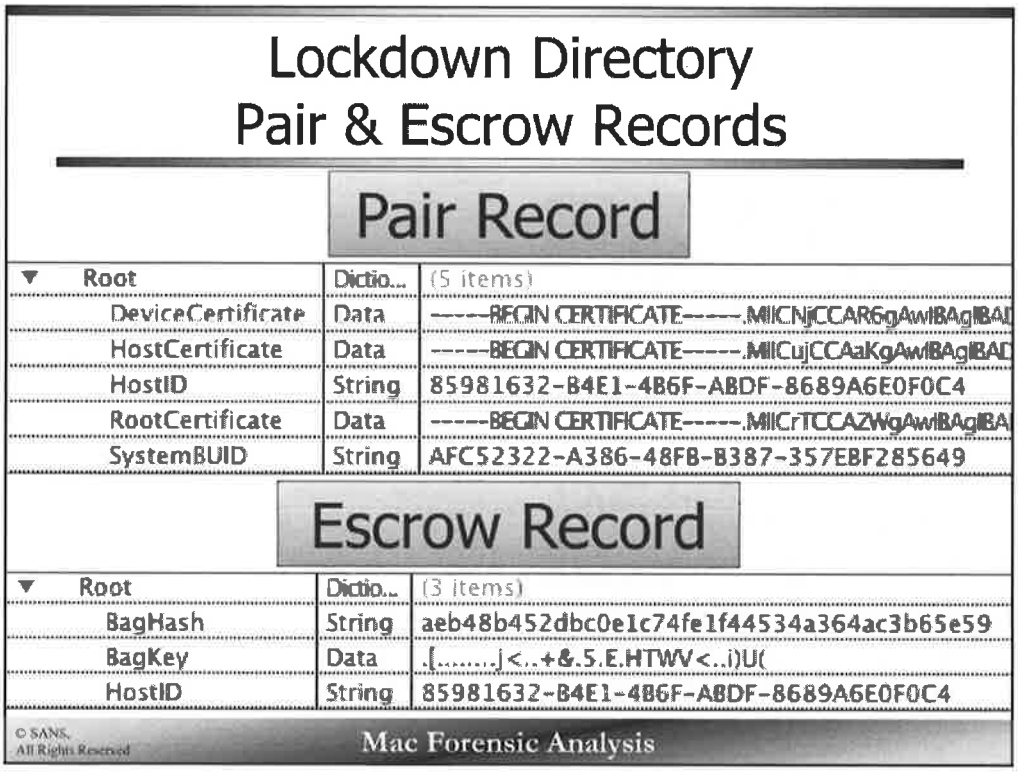
On live devices and physical images, the original GUID is kept, so the forensic examiner will need to review the contents of the GUID directory or the `com.apple.lsdidentifiers.plist` to determine which GUID is associated with a particular application.



When a device is plugged into a computer, it attempts to create a trust relationship with that system. On iOS 7 and newer devices a popup window will open allowing the user to select to "`Trust`" or "`Don't Trust`" this system (shown above). If the user selects "`Trust`" paired lockdown records will be created on both the system and the device.

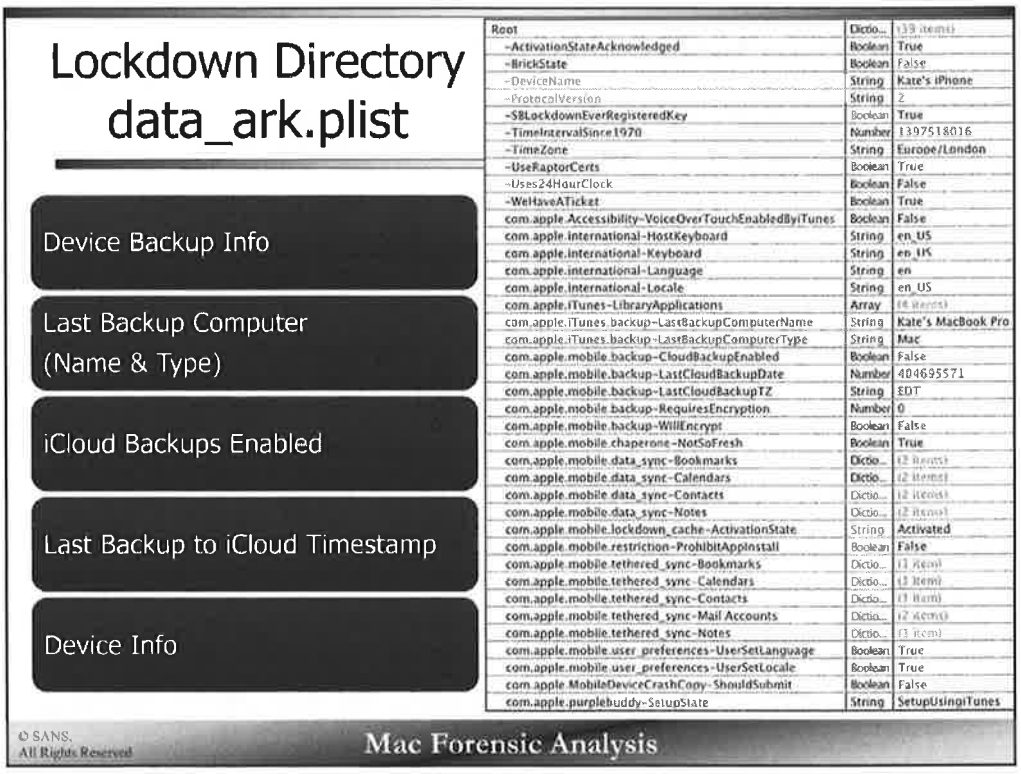
The lockdown records on the device are stored in the `/root/Library/Lockdown/` directory. Depending on how many systems this device has trusted you will have a record in each the `escrow_records` and `pair_records` that are named with the same GUID.


root
.obliterated
Library
Caches
com.apple.itunesstored
Lockdown
activation_records
wildcard_record.plist
data_ark.plist
device_private_key.pem
device_public_key.pem
escrow_records
80D8B54B-76BF-4CF5-A8AA-830BF394165D.plist
B3BE09CD-2952-47BF-91FC-F793D08203ED.plist
pair_records
80D8B54B-76BF-4CF5-A8AA-830BF394165D.plist
B3BE09CD-2952-47BF-91FC-F793D08203ED.plist



The lockdown records contains various bits of information such as escrow keys and certificates. One possible key may be able to tell us what system this device was trusted with – the `SystemBUID` ley in the `pair_records` property list files. This GUID matches the GUID found in a system's `SystemConfiguration.plist` file located in `/private/var/db/lockdown` directory.

If provided an iDevice along with a computer system, we can determine if these devices are trusted devices.



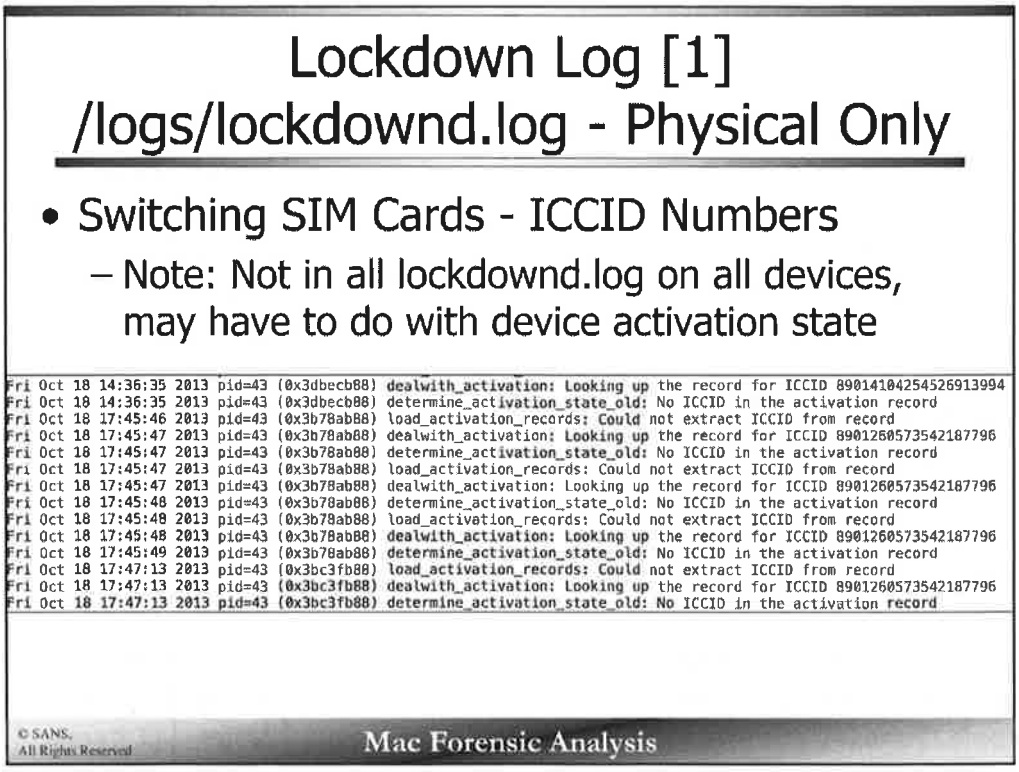
The `data_ark.plist` file located in the `/root/Library/Lockdown` directory contains information related to how the device is set to backup.

- What is the computer name and type of the last time this device was backed up.
- Are iCloud backups enabled, and when was the last iCloud backup.
- Do tethered backups require encryption.

This property list also contains device information such as:

- Device Name
- International Setup Information for Language, Keyboard, etc.
- iTunes App Library Contents

Root	Dictio...	(39 items)
-ActivationStateAcknowledged	Boolean	True
-BrickState	Boolean	False
-DeviceName	String	Kate's iPhone
-ProtocolVersion	String	2
-SBLockdownEverRegisteredKey	Boolean	True
-TimeIntervalSince1970	Number	1397518016
-TimeZone	String	Europe/London
-UseRaptorCerts	Boolean	True
-Uses24HourClock	Boolean	False
-WeHaveATicket	Boolean	True
com.apple.Accessibility-VoiceOverTouchEnabledByiTunes	Boolean	False
com.apple.international-HostKeyboard	String	en_US
com.apple.international-Keyboard	String	en_US
com.apple.international-Language	String	en
com.apple.international-Locale	String	en_US
com.apple.iTunes-LibraryApplications	Array	(4 items)
com.apple.iTunes.backup-LastBackupComputerName	String	Kate's MacBook Pro
com.apple.iTunes.backup-LastBackupComputerType	String	Mac
com.apple.mobile.backup-CloudBackupEnabled	Boolean	False
com.apple.mobile.backup-LastCloudBackupDate	Number	404695571
com.apple.mobile.backup-LastCloudBackupTZ	String	EDT
com.apple.mobile.backup-RequiresEncryption	Number	0
com.apple.mobile.backup-WillEncrypt	Boolean	False
com.apple.mobile.chaperone-NotSoFresh	Boolean	True
com.apple.mobile.data_sync-Bookmarks	Dictio...	(2 items)
com.apple.mobile.data_sync-Calendars	Dictio...	(2 items)
com.apple.mobile.data_sync-Contacts	Dictio...	(2 items)
com.apple.mobile.data_sync-Notes	Dictio...	(2 items)
com.apple.mobile.lockdown_cache-ActivationState	String	Activated
com.apple.mobile.restriction-ProhibitAppInstall	Boolean	False
com.apple.mobile.tethered_sync-Bookmarks	Dictio...	(1 item)
com.apple.mobile.tethered_sync-Calendars	Dictio...	(1 item)
com.apple.mobile.tethered_sync-Contacts	Dictio...	(1 item)
com.apple.mobile.tethered_sync-Mail Accounts	Dictio...	(2 items)
com.apple.mobile.tethered_sync-Notes	Dictio...	(1 item)
com.apple.mobile.user_preferences-UserSetLanguage	Boolean	True
com.apple.mobile.user_preferences-UserSetLocale	Boolean	True
com.apple.MobileDeviceCrashCopy-ShouldSubmit	Boolean	False
com.apple.purplebuddy-SetupState	String	SetupUsingiTunes

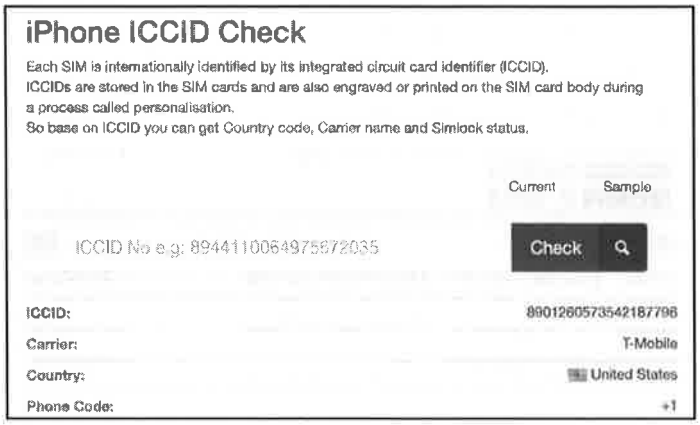


The `lockdownd` daemon also writes a log found in `/logs/lockdownd.log`. This long may contain a historical view of various SIM cards inserted into the system. While not in all `lockdownd.log` files, this information may come in handy if attempting to determine if the user was travelling or using different phone numbers.

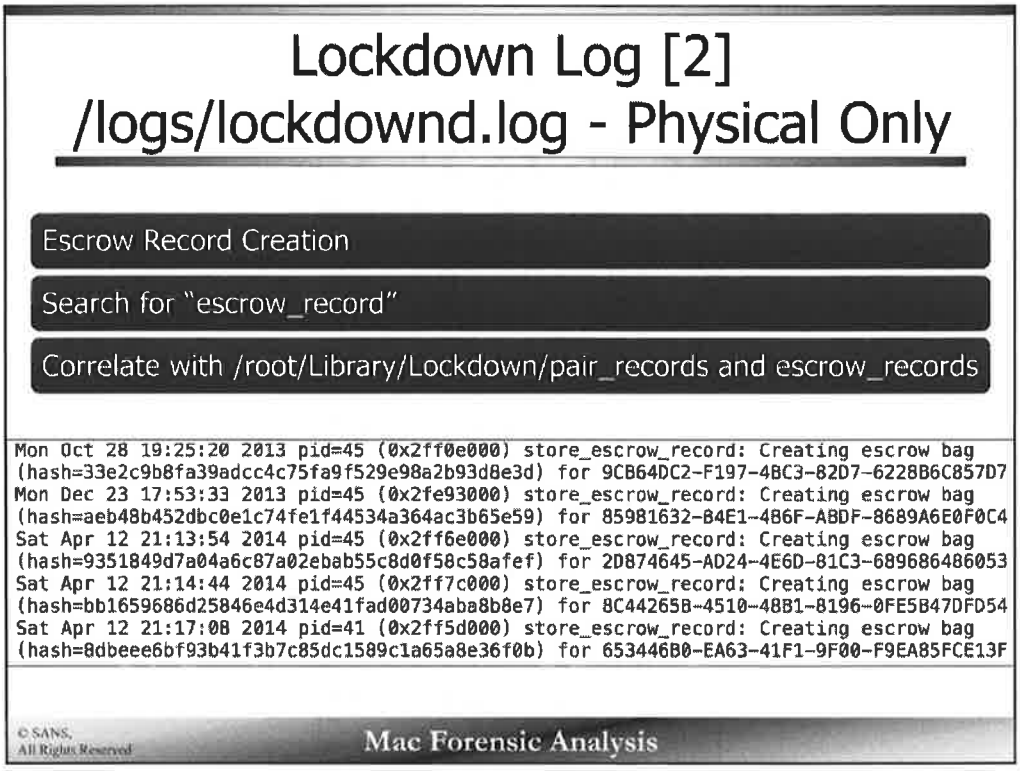
This example is from an iPhone 3GS, running iOS 6.1.3, an iPhone 4S running the same OS did not have these records. More research is needed to determine the reason why.

The example above shows two SIM cards were inserted into this phone each on October 18th, 2013. One at 14:36 and another at 17:45.

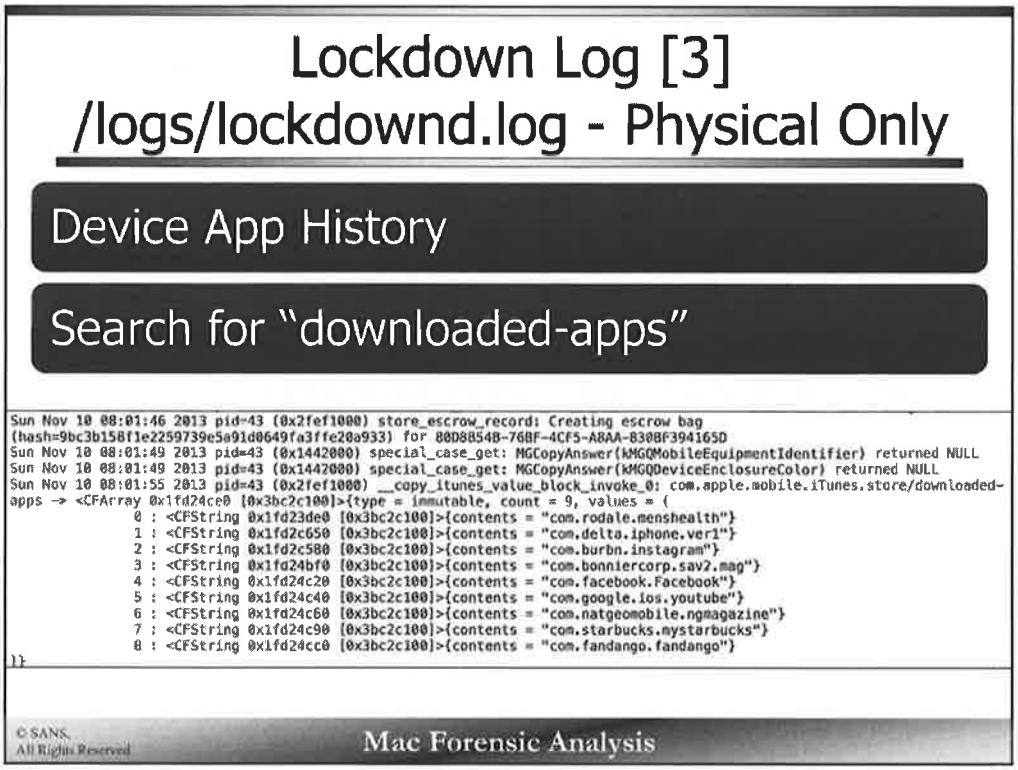
The ICCID numbers can be put into a checker (https://imeidata.net/iphone/iccid-check) to determine country and carrier information.



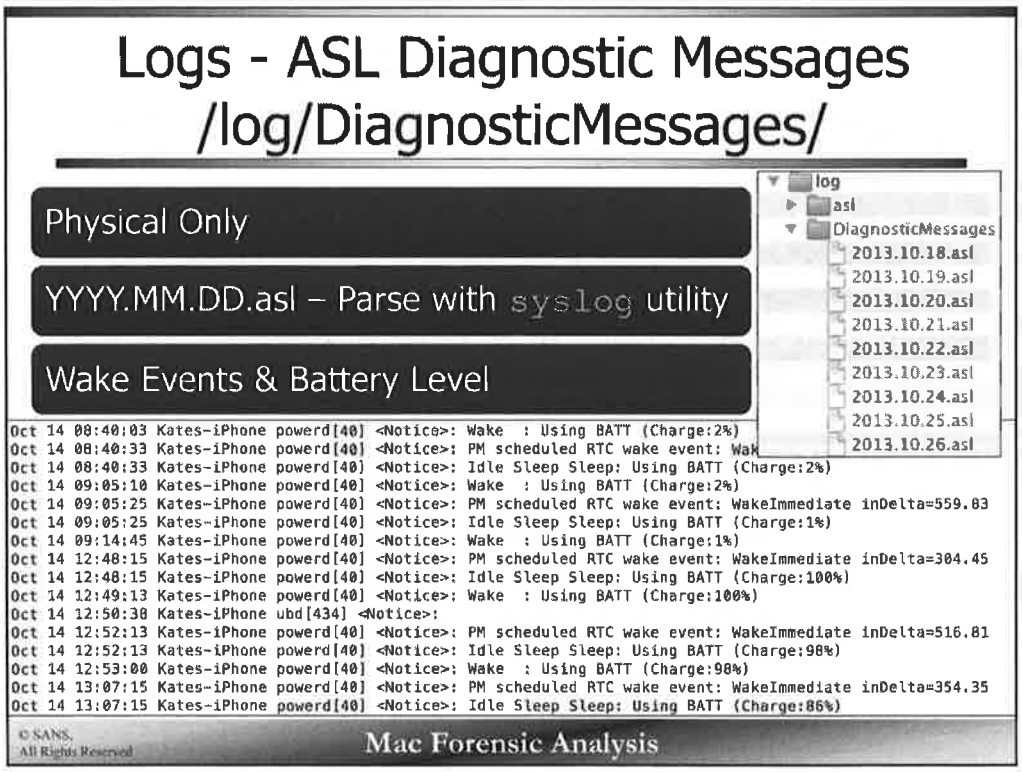
```
Fri Oct 18 14:36:35 2013 pid=43 (0x3dbecb88) dealwith_activation: Looking up the record for ICCID 89014104254526913994
Fri Oct 18 14:36:35 2013 pid=43 (0x3dbecb88) determine_activation_state_old: No ICCID in the activation record
Fri Oct 18 17:45:46 2013 pid=43 (0x3b78ab88) load_activation_records: Could not extract ICCID from record
Fri Oct 18 17:45:47 2013 pid=43 (0x3b78ab88) dealwith_activation: Looking up the record for ICCID 8901260573542187796
Fri Oct 18 17:45:47 2013 pid=43 (0x3b78ab88) determine_activation_state_old: No ICCID in the activation record
Fri Oct 18 17:45:47 2013 pid=43 (0x3b78ab88) load_activation_records: Could not extract ICCID from record
Fri Oct 18 17:45:47 2013 pid=43 (0x3b78ab88) dealwith_activation: Looking up the record for ICCID 8901260573542187796
Fri Oct 18 17:45:48 2013 pid=43 (0x3b78ab88) determine_activation_state_old: No ICCID in the activation record
Fri Oct 18 17:45:48 2013 pid=43 (0x3b78ab88) load_activation_records: Could not extract ICCID from record
Fri Oct 18 17:45:48 2013 pid=43 (0x3b78ab88) dealwith_activation: Looking up the record for ICCID 8901260573542187796
Fri Oct 18 17:45:49 2013 pid=43 (0x3b78ab88) determine_activation_state_old: No ICCID in the activation record
Fri Oct 18 17:47:13 2013 pid=43 (0x3bc3fb88) load_activation_records: Could not extract ICCID from record
Fri Oct 18 17:47:13 2013 pid=43 (0x3bc3fb88) dealwith_activation: Looking up the record for ICCID 8901260573542187796
Fri Oct 18 17:47:13 2013 pid=43 (0x3bc3fb88) determine_activation_state_old: No ICCID in the activation record
```



The `lockdownd.log` also contains timestamped records of when that trust relationship was created. Every time a device pairs with a system a `store_escrow_record` gets created with the associated GUID. This is a nice way to corroborate when these records were created.



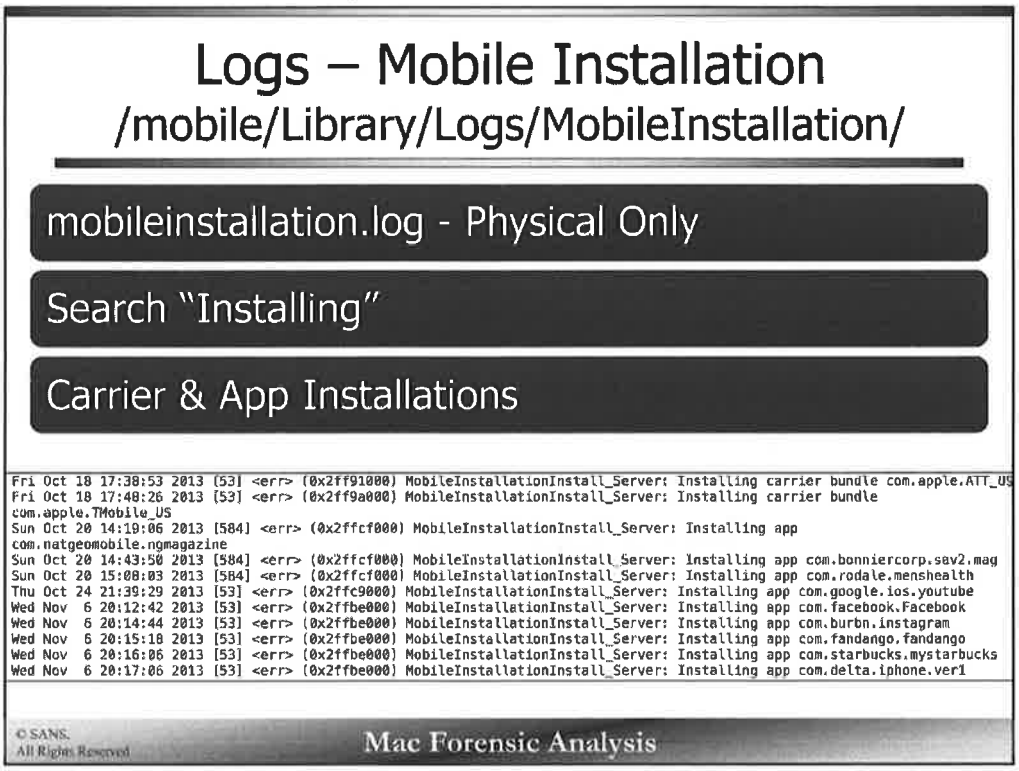
The `lockdownd.log` also provides some insight on apps installed on the device. If an app has been removed, it may be worth looking in this log file for remnants of its existence. When a device is connected to a system it wants to copy the app files to the system, so it lists the files installed on the device in the `__copy_itune_value_block_invoke_0` records.



Located in the `/log/DiagnosticMessages/` directory on a physical image, ASL logs are named with a standard format similar to those on OS X - `YYYY.MM.DD.asl`.

iDevices have ASL logs as well, while not as detailed as those on OS X we can still gather some information on them such when the device was on and what the battery level was.

These logs can be extracted and parsed by the same tool used on OS X, `syslog`.



The `mobileinstallation.log` located in the `/mobile/Library/Logs/MobileInstallation/` directory contains a historical view of when apps were installed and carrier bundles were installed.

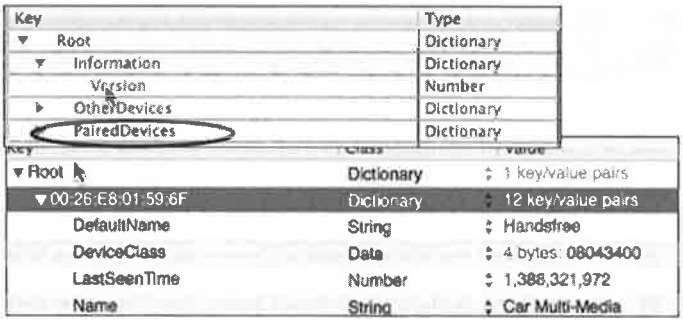
Similar to the `lockdownd.log` file we can see a change in networks, from AT&T to T-Mobile when carrier bundles were updated.

When an app is installed, it is recorded in this log using the application bundle identifier (reverse DNS format name). For example, Instagram (`com.burbn.instagram`) was installed on November 6th, 2013 at 20:14.

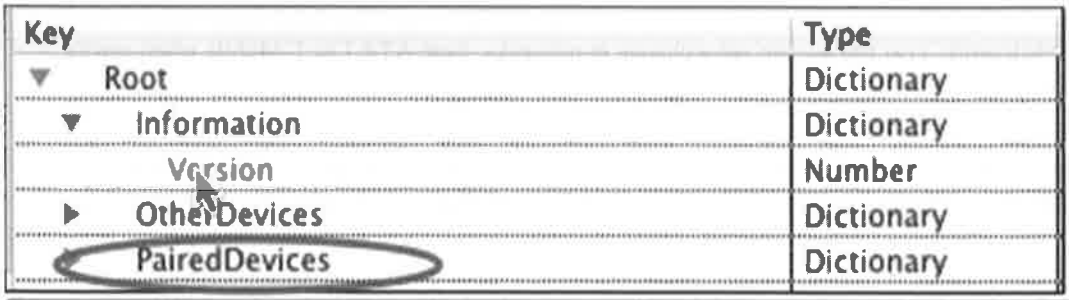
Bluetooth Devices

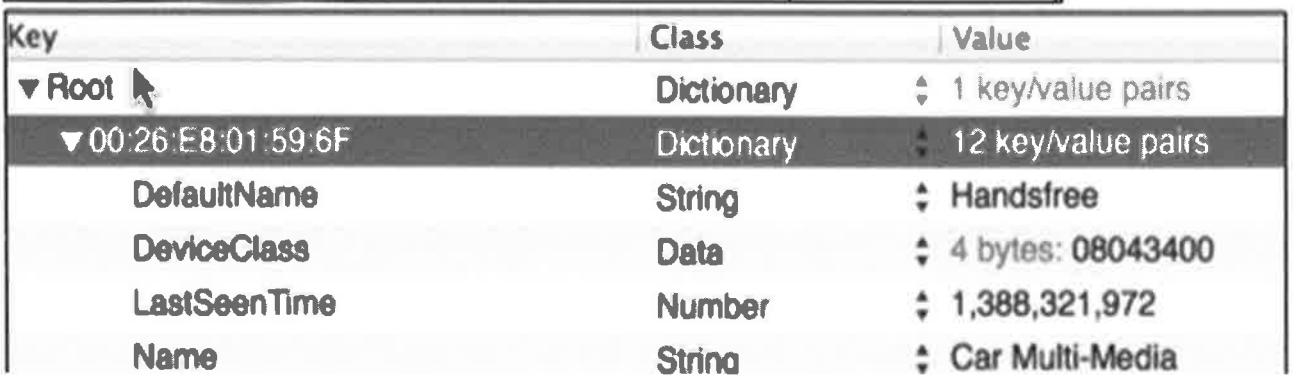
com.apple.MobileBluetooth.ledevices.plist

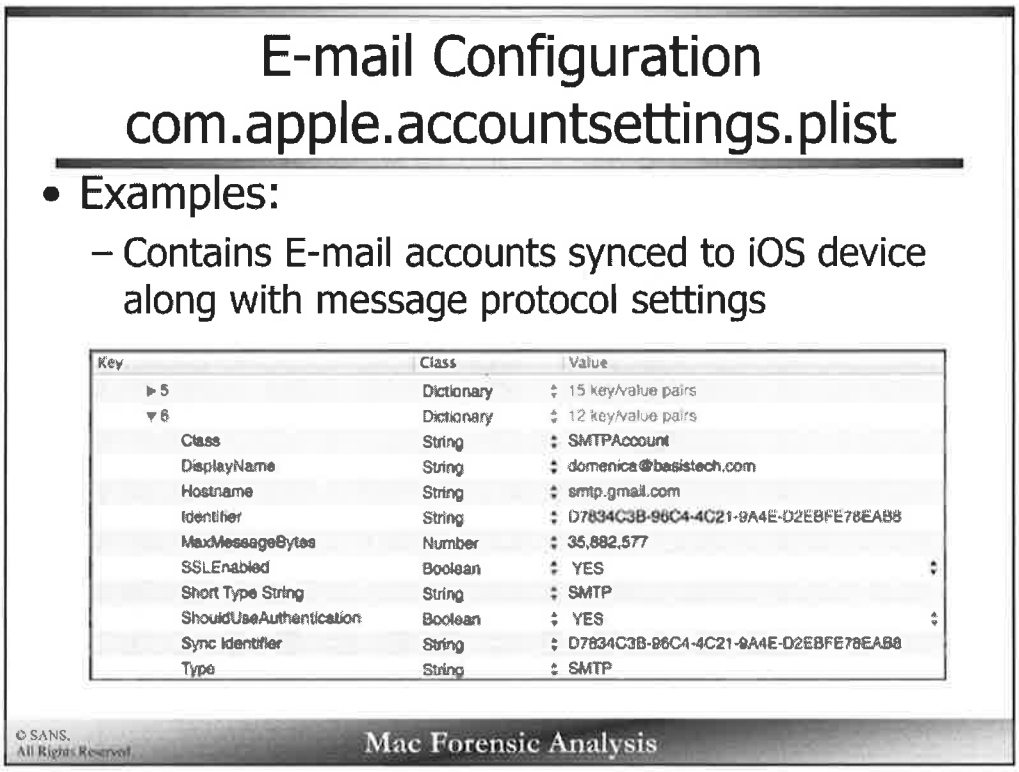
- Examples:
  - Contains information (BT Address for devices synced with iOS device



© SANS
All Rights Reserved



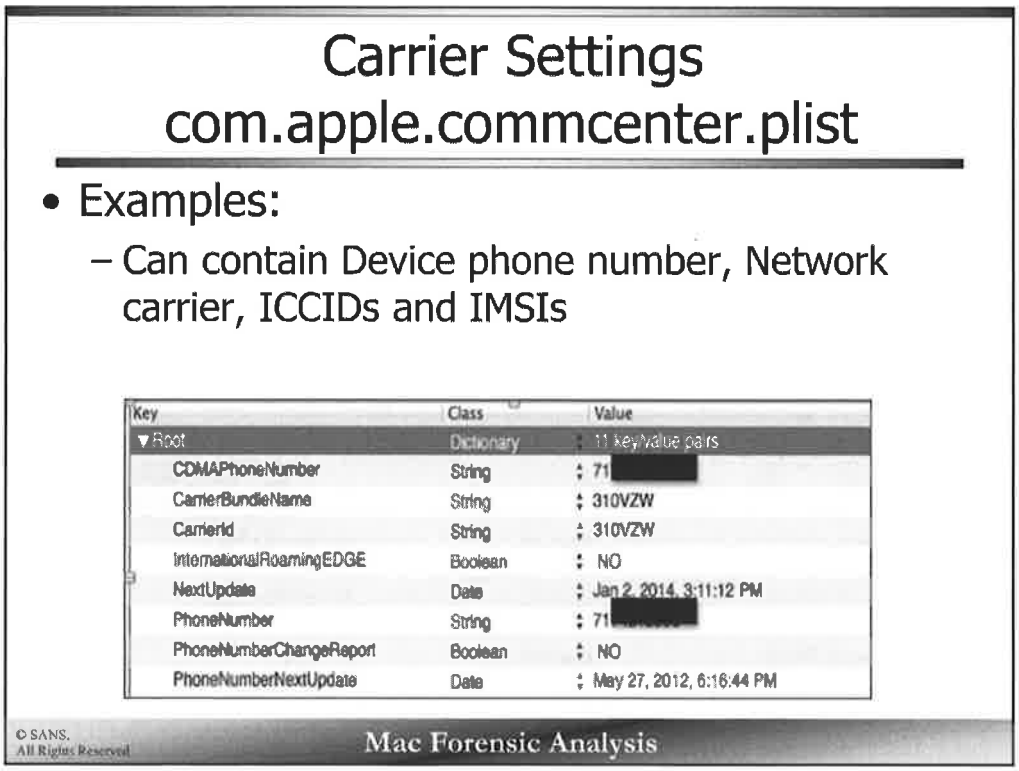




Each E-mail account that was configured to receive e-mail on the iOS device will maintain an entry in the com.apple.accountsettings.plist file. In addition to containing the e-mail address, settings like message protocol and maximum allowable bytes per message are stored in the plist file. Some other preference plist files that can contain messaging account information include:

- Library/Preferences/com.apple.imservice
- Library/Preferences/com.apple.madrid.plist (iOS 5)
- Library/Preferences/com.apple.MobileSMS.plist
- Library/Preferences/com.apple.gamed.plist
- Library/Preferences/com.apple.homesharing.plist
- Library/Preferences/com.apple.imserviceFaceTime.plist
- Library/Preferences/com.apple.imserviceiMessage.plist
- Library/Preferences/com.apple.MailAccount-ExtProperties.plist

Key	Class	Value
▶ 5	Dictionary	15 key/value pairs
▼ 6	Dictionary	12 key/value pairs
Class	String	SMTPAccount
DisplayName	String	domenica@basistech.com
Hostname	String	smtp.gmail.com
Identifier	String	D7834C3B-96C4-4C21-9A4E-D2EBFE78EAB8
MaxMessageBytes	Number	35,882,577
SSLEnabled	Boolean	YES
Short Type String	String	SMTP
ShouldUseAuthentication	Boolean	YES
Sync Identifier	String	D7834C3B-96C4-4C21-9A4E-D2EBFE78EAB8
Type	String	SMTP

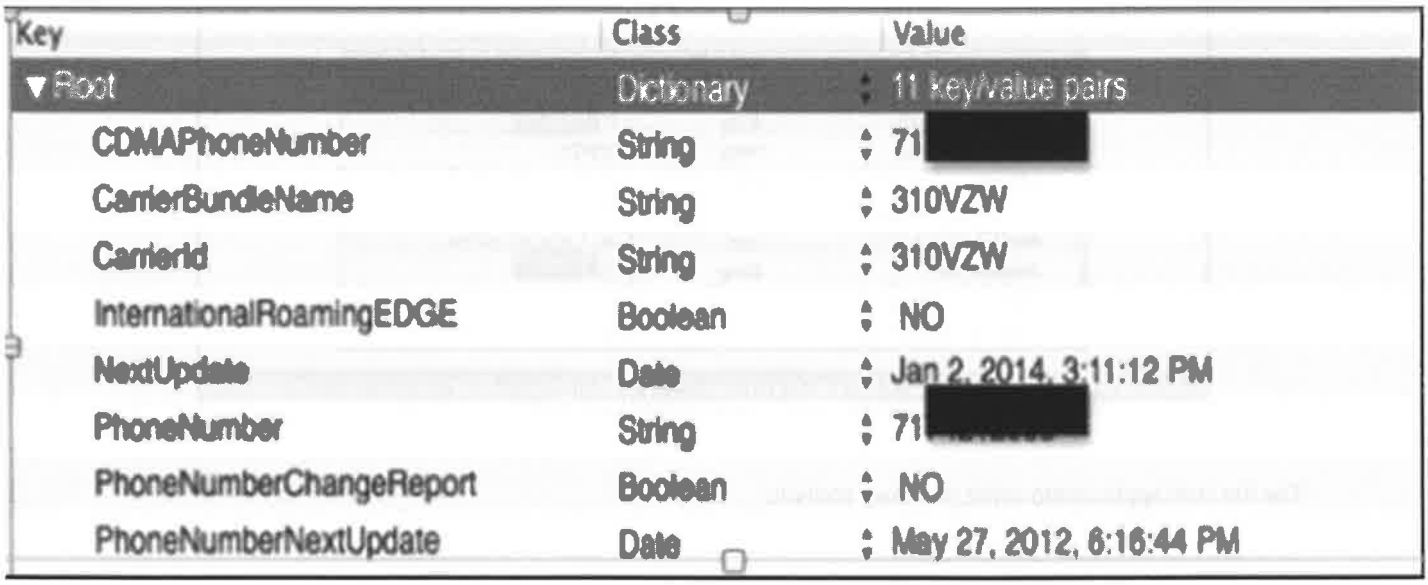


The file com.apple.commcenter.plist may contain:

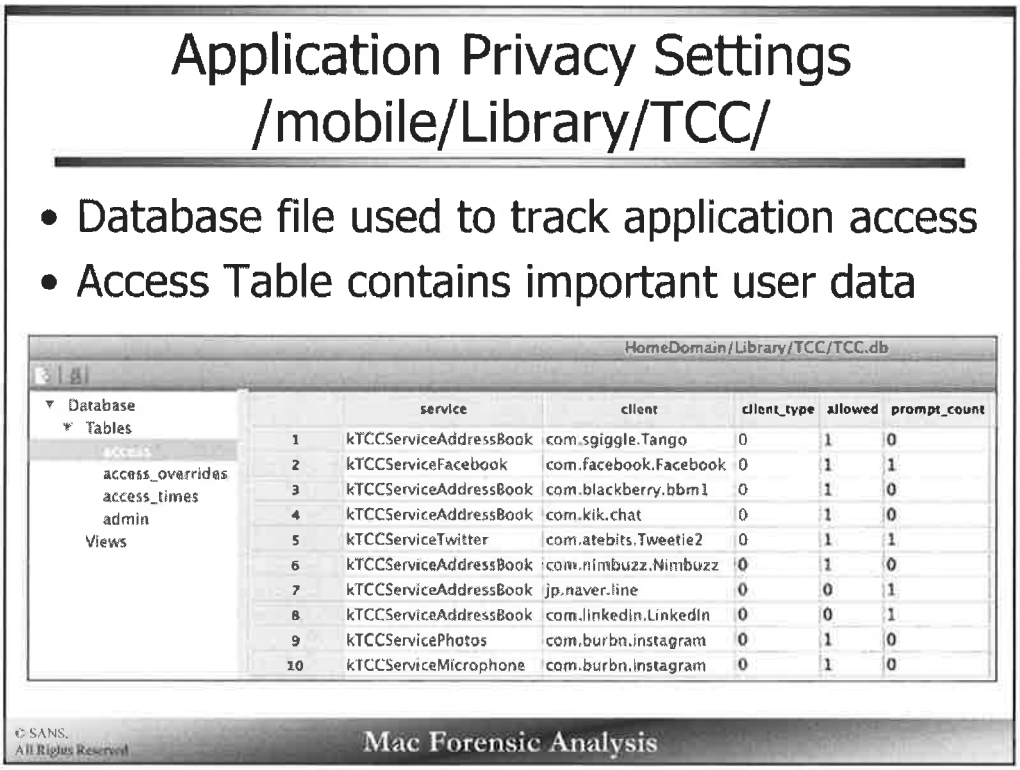
- Device phone number
- Network Carrier
- ICCID
- IMSI

Some other preference files containing similar information include:

- Library/Preferences/com.apple.preferences.network
- Library/Preferences/com.apple.network.eapclient.tls.TrustExceptions.plist



Key	Class	Value
▼Root	Dictionary	11 key/value pairs
CDMAPhoneNumber	String	71[redacted]
CarrierBundleName	String	310VZW
CarrierId	String	310VZW
InternationalRoamingEDGE	Boolean	NO
NextUpdate	Date	Jan 2, 2014, 3:11:12 PM
PhoneNumber	String	71[redacted]
PhoneNumberChangeReport	Boolean	NO
PhoneNumberNextUpdate	Date	May 27, 2012, 6:16:44 PM



`TCC.db` is a SQLite database stored in the location: `/mobile/Library/TCC`. This database logs the accesses that are granted to various installed applications.

These settings can be configured by the user for applications installed on the device by accessing: Settings > Privacy from the device menu.

The following items are configurable and applications can be granted access to:

- Contacts
- Calendars
- Reminders
- Photos
- Bluetooth Sharing
- Microphone
- Motion Activity

It is also possible to leverage the social account data residing in some applications by third-party applications. These will appear below the system accesses. Examples shown above:

- `KTCCServiceTwitter`
- `KTCCServiceFacebook`

The table in the SQLite database of most importance is "Access"
This table has several columns but of particular importance are:

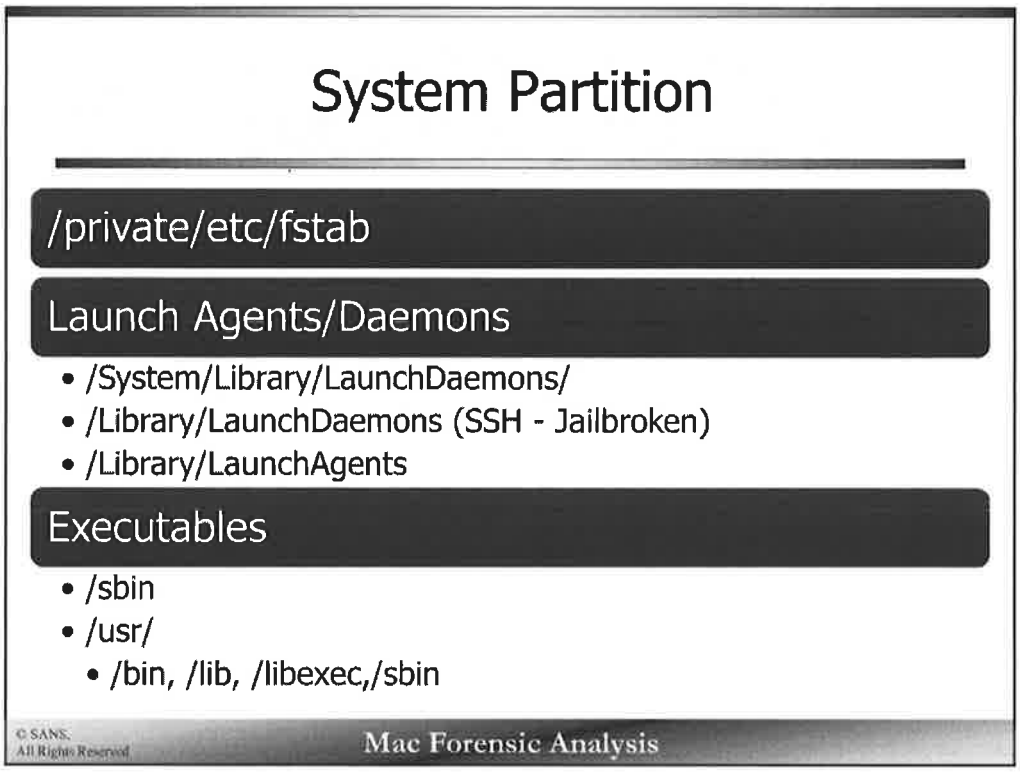
- Service - lists the KTTCService that is being accessed by the application (ex: `KTCCAddressBook`)
- Client – refers to the Application requesting the service (ex. Facebook)
- Allowed – this column will show the services currently requesting the service. 1 = Allowed, 0 = Not Allowed. If the service was never requested, there will be no entry in this table. Once access is denied for an existing application, the Allowed column will turn to zero "0".

HomeDomain/Library/TCC/TCC.db

- Database
  - Tables
    - access
    - access_overrides
    - access_times
    - admin
  - Views

	service	client	client_type	allowed	prompt_count
1	kTCCServiceAddressBook	com.sgiggle.Tango	0	1	0
2	kTCCServiceFacebook	com.facebook.Facebook	0	1	1
3	kTCCServiceAddressBook	com.blackberry.bbm1	0	1	0
4	kTCCServiceAddressBook	com.kik.chat	0	1	0
5	kTCCServiceTwitter	com.atebits.Tweetie2	0	1	1
6	kTCCServiceAddressBook	com.nimbuzz.Nimbuzz	0	1	0
7	kTCCServiceAddressBook	jp.naver.line	0	0	1
8	kTCCServiceAddressBook	com.linkedin.LinkedIn	0	0	1
9	kTCCServicePhotos	com.burbn.instagram	0	1	0
10	kTCCServiceMicrophone	com.burbn.instagram	0	1	0

110

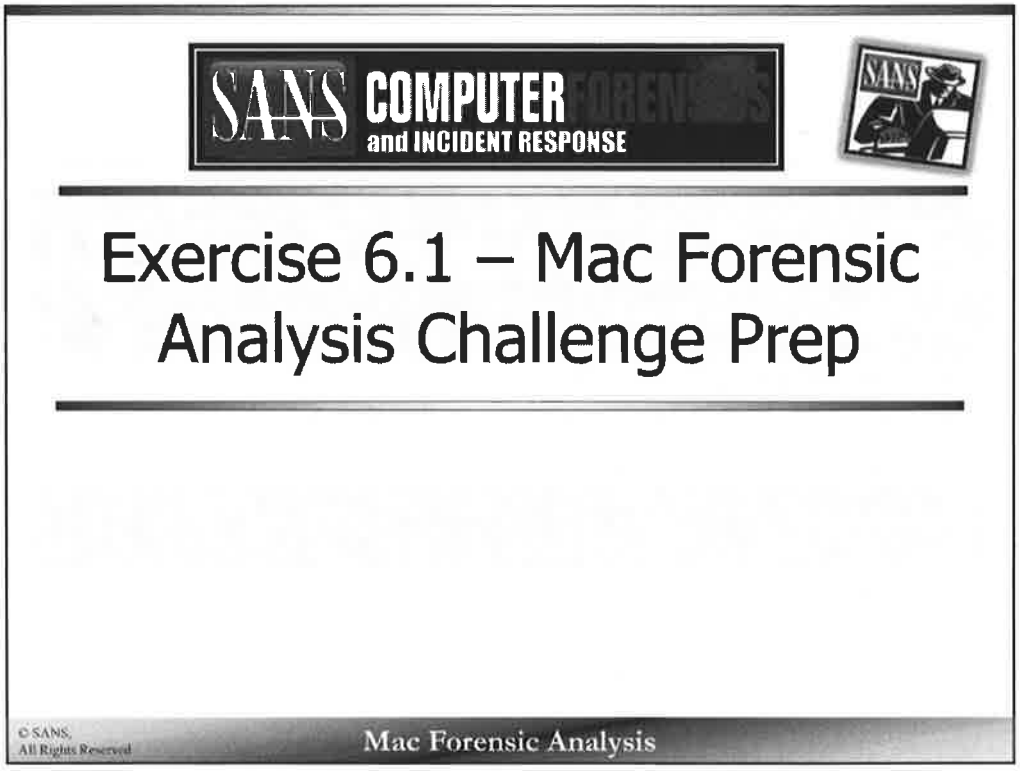


While there is not a great amount of evidentiary data on the `/System` partition it might behoove you to look here for certain cases where tampering or malware may play a part in the investigation.

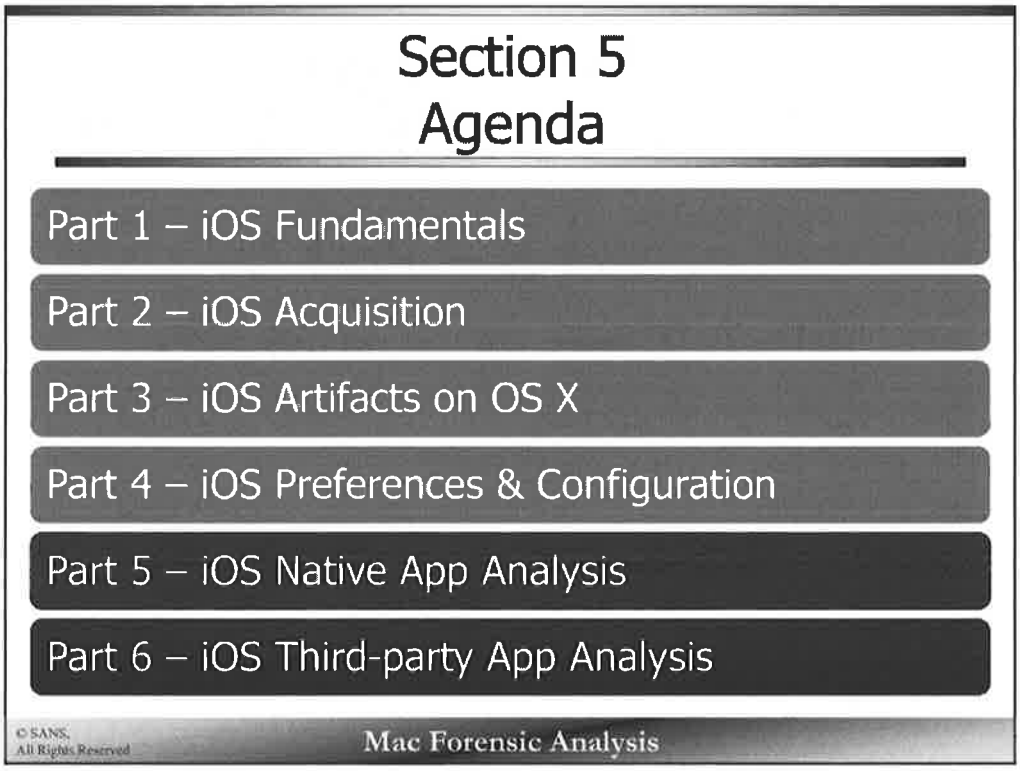
The file system table at `/private/etc/fstab` may show you if a device has been jailbroken as show in previous slides.

Launch Agents and Daemons may provide insight into what process are running – these areas are particularly interesting on jailbroken devices where various programs are launched from here such as SSH.

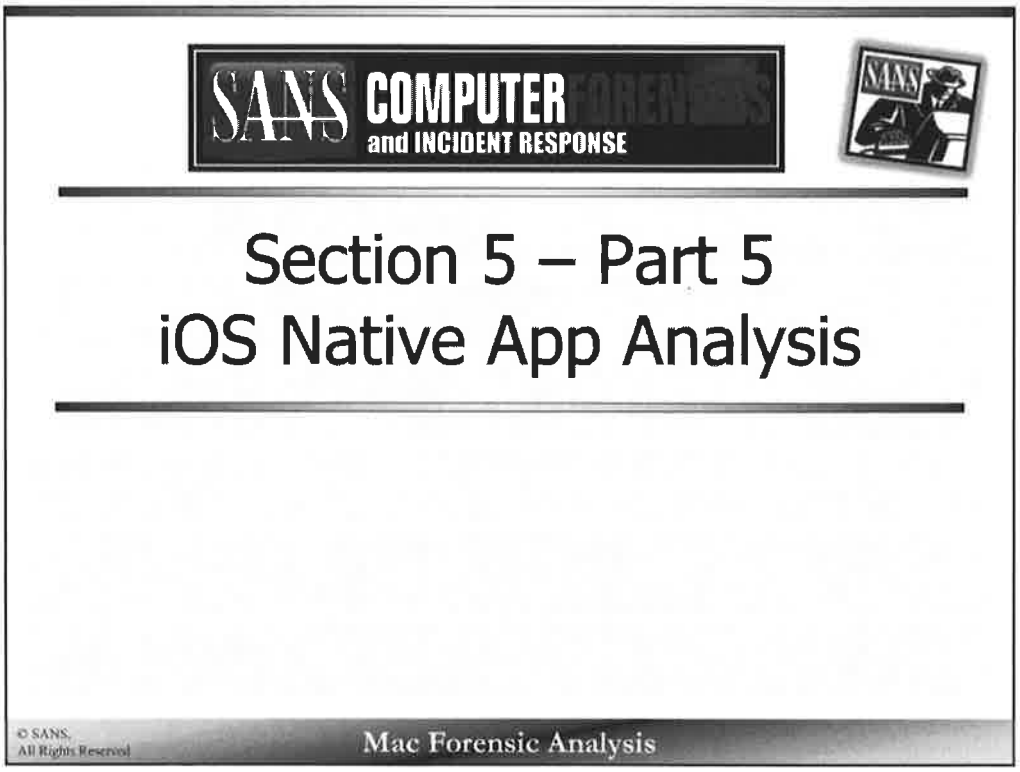
A thorough look at executables may determine if system executables or malware was installed on the device. Compare these with known good samples on a similar OS.



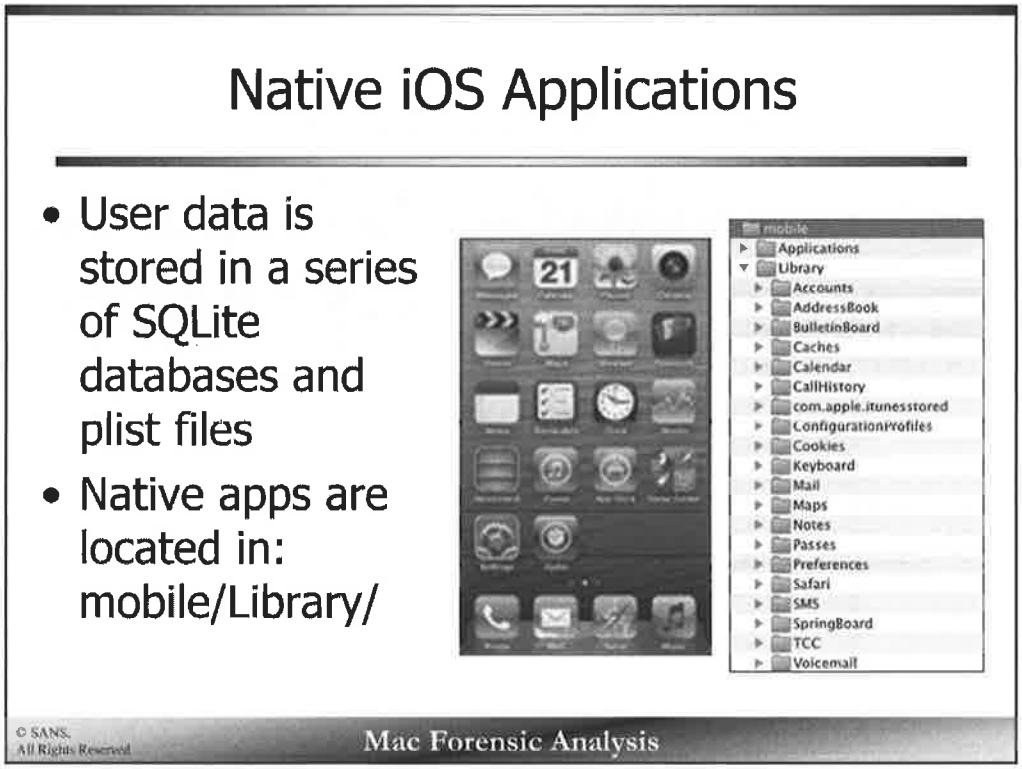
This page intentionally left blank.



This page intentionally left blank.



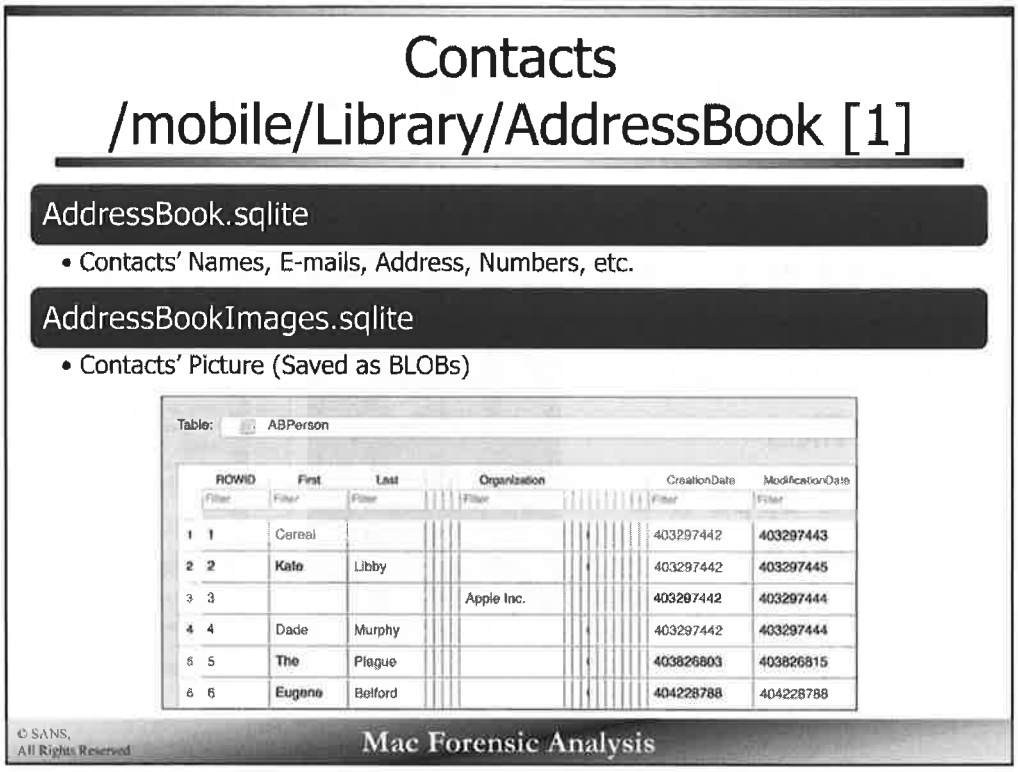
This page intentionally left blank.



iOS device ship with several "Native" iOS applications that have been pre-configured. These Native applications will vary slightly based on the iOS firmware version. Included in this list are many of the applications that are present on the home screen without downloading and installing from iTunes.

You will find this same data on each iOS device located in the same respective folders even if the data in the UI has been moved around. The native iOS applications contain the core data that is often accessed by other native applications as well as third-party applications downloaded from iTunes through the use of Apple's APIs. The Address Book may often be considered the most interactive database, because its content is often widely shared with other applications. It is not uncommon to see duplicate data shared throughout multiple third-party applications.

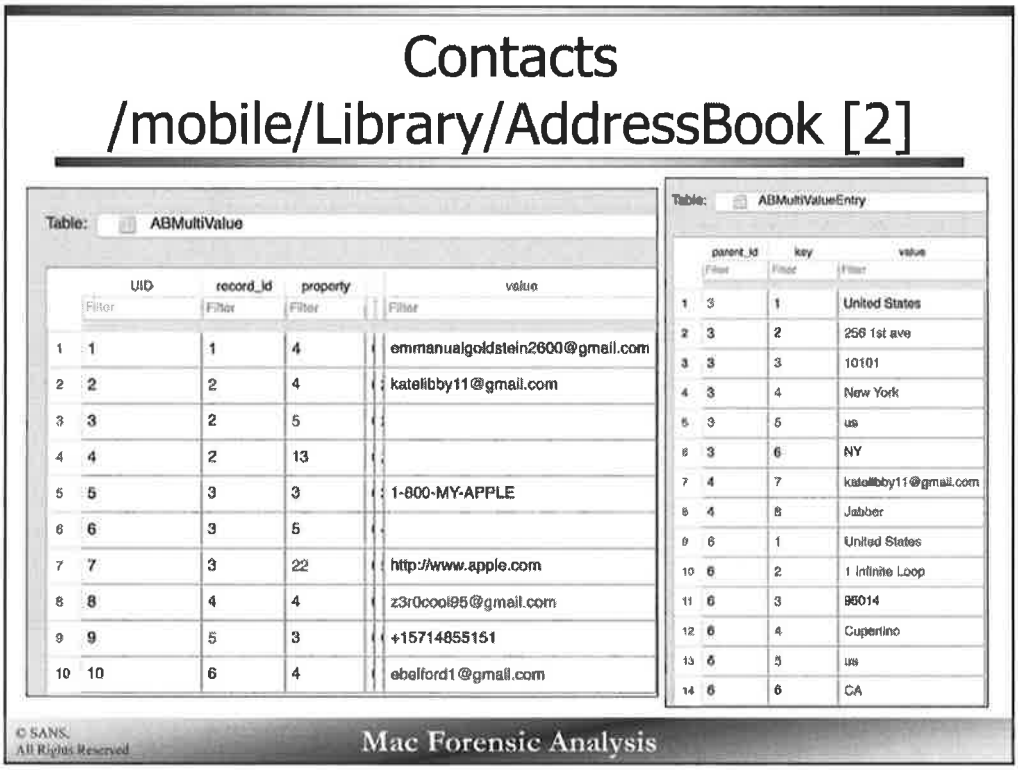
The applications that most commonly contain user data of interest will be discussed in further detail.



The Contacts application stores its data in two SQLite databases, `AddressBook.sqlite` and `AddressBookImages.sqlite`.

The `AddressBook.sqlite` database contains the contacts information including Names, e-mails, addresses, phone numbers, social media accounts, etc. The `AddressBookImages.sqlite` database only holds the contacts' associated image in a BLOB in the 'data' column of the `ABFullSizeImage` table.

The `ABPerson` table in `AddressBook.sqlite` stores the contacts names, organization, job title, etc. This table also stores the contact creation and modification dates.



The `ABMultiValue` and `ABMultiValueKey` table sin `AddressBook.sqlite` stores other metadata associated with contacts such as e-mail address, usernames, phone numbers and addresses.

Each contact's information is stored under a different `record_id` or `parent_id` depending on the table. The information for `record_id=4` and `parent_id=4` are for the same contact.

Table: ABMultiValueEntry

	parent_id	key	value
	Filter	Filter	Filter
1	3	1	United States
2	3	2	256 1st ave
3	3	3	10101
4	3	4	New York
5	3	5	us
6	3	6	NY
7	4	7	katelibby11@gmail.com
8	4	8	Jabber
9	6	1	United States
10	6	2	1 Infinite Loop
11	6	3	95014
12	6	4	Cupertino
13	6	5	us
14	6	6	CA

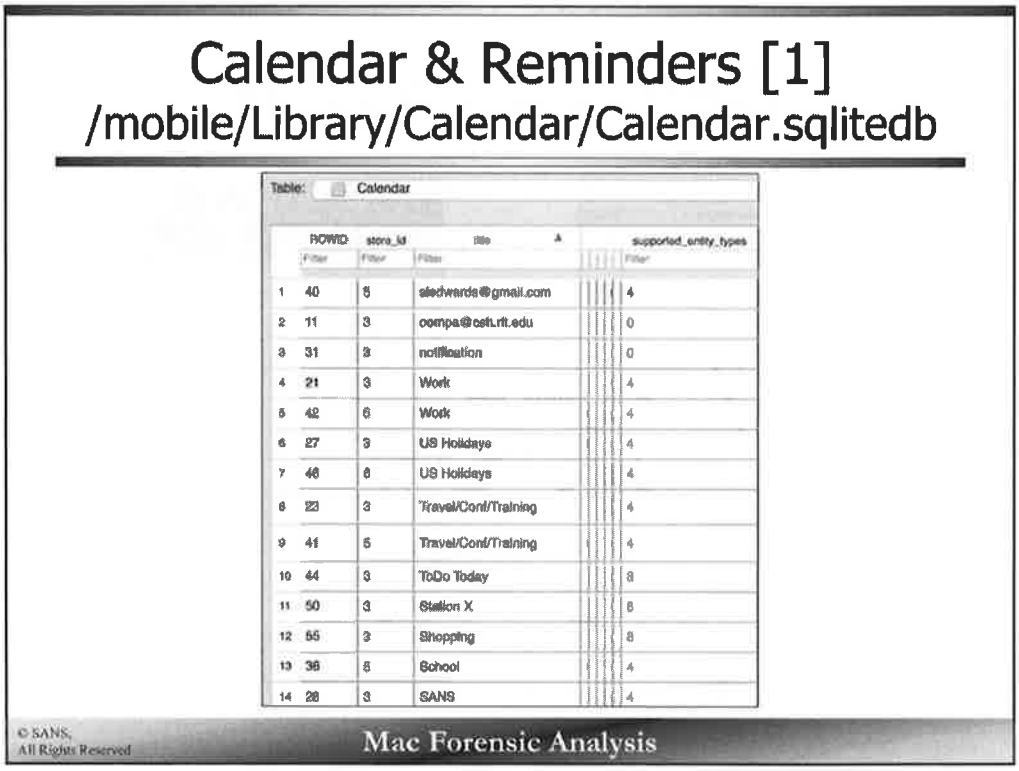
Table: ABMultiValue

	UID	record_id	property	value
	Filter	Filter	Filter	Filter
1	1	1	4	emmanualgoldstein2600@gmail.com
2	2	2	4	katelibby11@gmail.com
3	3	2	5	
4	4	2	13	
5	5	3	3	1-800-MY-APPLE
6	6	3	5	
7	7	3	22	http://www.apple.com
8	8	4	4	z3r0cool95@gmail.com
9	9	5	3	+15714855151
10	10	6	4	ebelford1@gmail.com

The Calendar and Reminders applications use the same database to store their contents.

These items can be synced from a variety of accounts including iCloud and Google as shown above in the left screenshot. It can include both personal calendars and shared calendars.

The Reminders application stores lists created by the user that may have deadlines and other information associated with each reminder item.

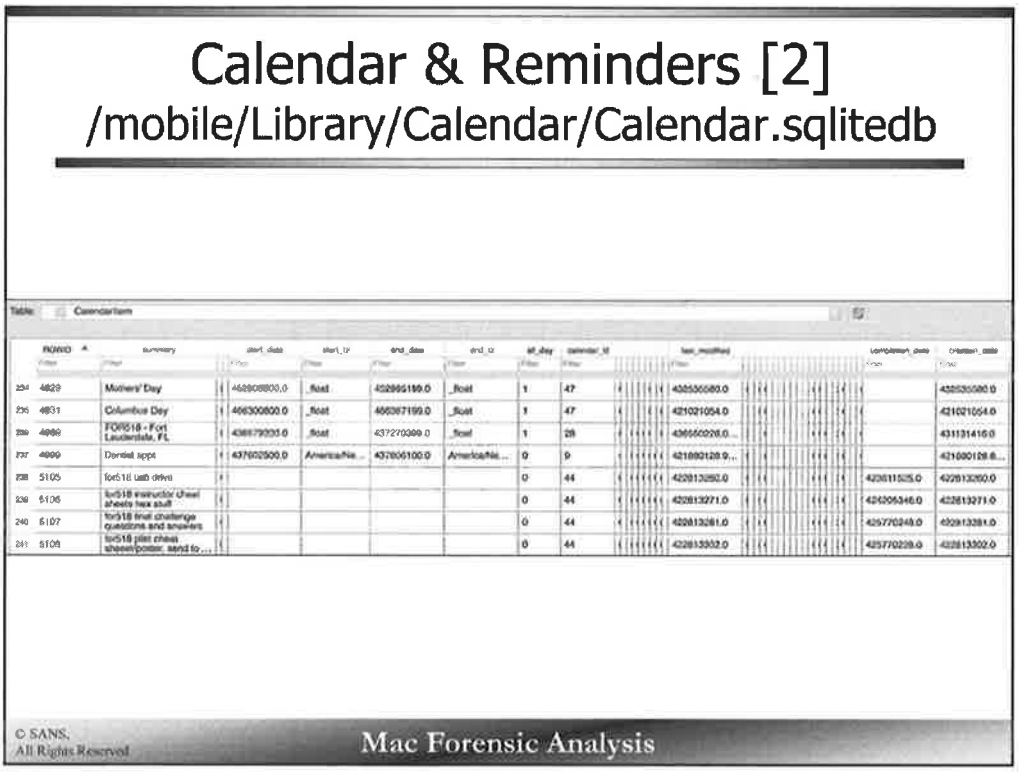


The `Calendar.sqlitedb` database located in /mobile/Library/Calendar contains the Calendar information for the device.

The Calendar table show shows contains information about which calendars are setup on the device. This may include local calendars, those synced with iCloud, Google calendars, and Reminders that are used with the Reminder application on the device or synced via iCloud.

Items that have the following `supported_entity_type`:

- 0 – iCloud
- 4 – Calendars
- 8 – Reminders



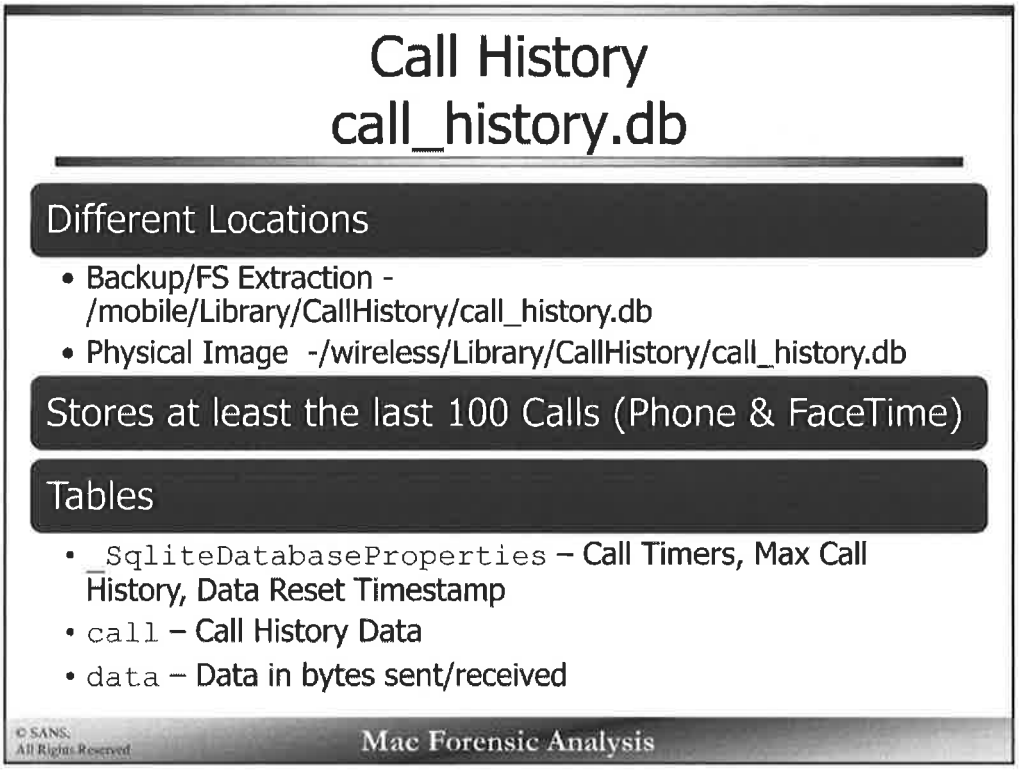
The `CalendarItem` table in the `Calendar.sqlitedb` database contains each calendar item or reminder with a variety of data, partially shown here for brevity.

Items included are:

- Summary/Item Name
- Start and End dates
- If the calendar items is an all-day event
- What calendar it belongs to (review the Calendar table)
- When the event or reminder was last modified, created
- When the reminder was completed
- …so much more!

Table: CalendarItem

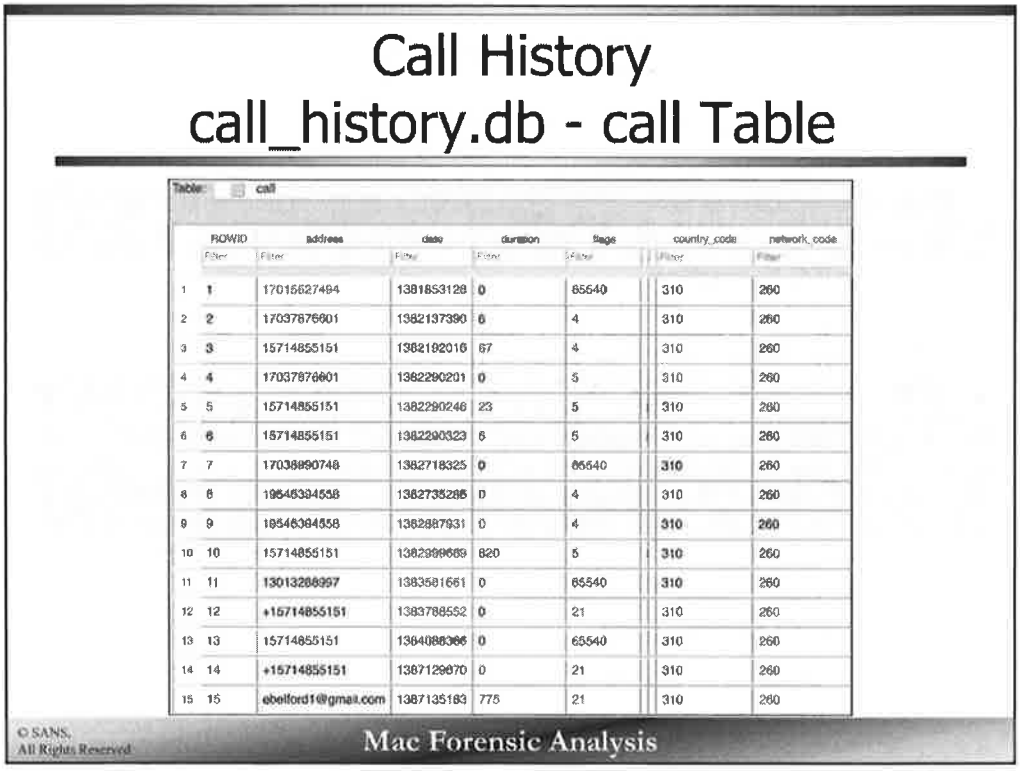
	ROWID	summary	start_date	start_tz	end_date	end_tz	all_day	calendar_id	last_modified	completion_date	creation_date
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
234	4929	Mothers' Day	452908800.0	_float	452995199.0	_float	1	47	432535580.0		432535580.0
235	4931	Columbus Day	466300800.0	_float	466387199.0	_float	1	47	421021054.0		421021054.0
236	4969	FOR518 - Fort Lauderdale, FL	436579200.0	_float	437270399.0	_float	1	28	436550226.0...		431131416.0
237	4999	Dentist appt	437602500.0	America/Ne...	437606100.0	America/Ne...	0	9	421860128.9...		421860128.8...
238	5105	for518 usb drive					0	44	422813260.0	423611525.0	422813260.0
239	5106	for518 instructor cheat sheets hex stuff					0	44	422813271.0	424205346.0	422813271.0
240	5107	for518 final challenge questions and answers					0	44	422813281.0	425770248.0	422813281.0
241	5108	for518 plist cheat sheeet/poster, send to ...					0	44	422813302.0	425770228.0	422813302.0



The call records may be in different locations depending on the type of acquisition. In backups and file system extractions, they are located in `/mobile/Library/CallHistory`. On a physical image, they are located in `/wireless/Library/CallHistory/`.

These records store at least the last 100 calls as shown in the `SqliteDatabaseProperties` table.

The call table contains the mail call records, while the data table shows the data that has been send or received by the device.



The call table in the `call_history.db` database contains a record for each call sent or received including FaceTime calls.

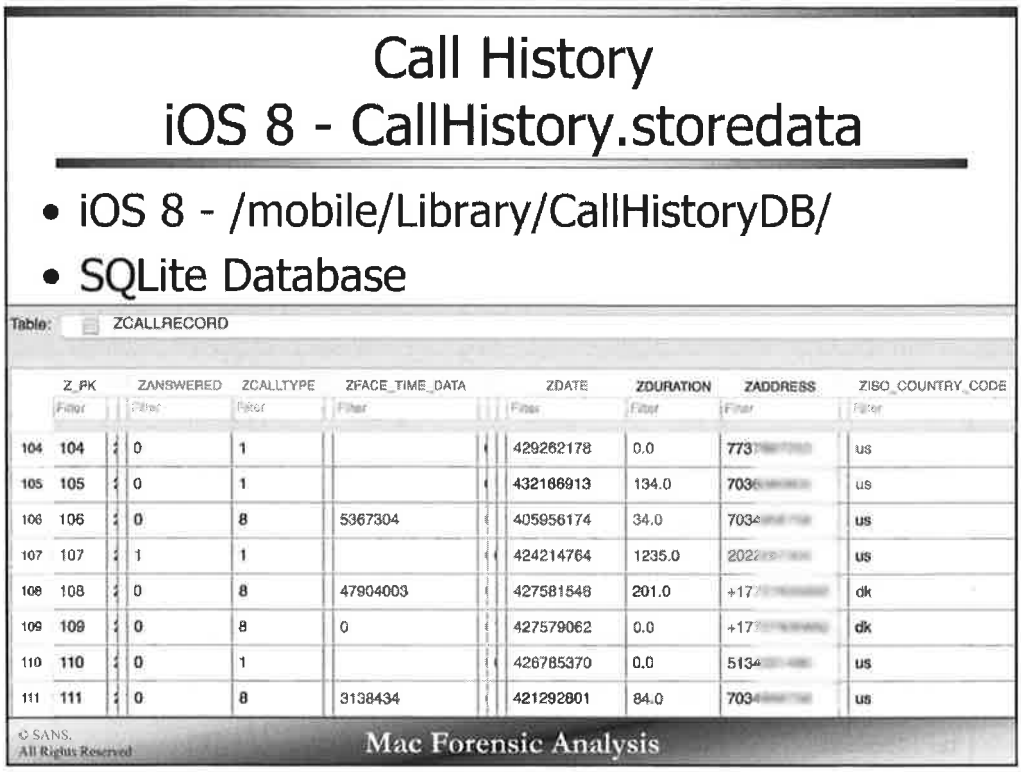
Each record contains the following data:

- `Address` – Phone number or e-mail of the contact.
- `Date` – When the call occurred.
- `Duration` – Time in seconds of the call.
- `Flags` – Direction/State of the call
  - iOS 6
    - 4 – Incoming
    - 5 - Outgoing
    - 65540 – Outgoing Canceled
    - 21 – Outgoing/Canceled FaceTime Call (If duration is 0)
  - iOS 7+ –
    - 0 – Incoming/Missed Call (if duration is 0)
    - 8 – Missed Incoming/Unknown Caller
    - 9 – Outgoing Call/Canceled Call (if duration is 0)
    - 16 – Canceled Incoming FaceTime
    - 17 – Missed FaceTime
    - 64 – Incoming FaceTime Audio/Missed FaceTime (If duration is 0)
    - 65 – Canceled Outgoing FaceTime
    - 65545 – Canceled Incoming
    - 1769472 – Missed Call
- Country & Network Codes

Correlate the address with the data found in the Address Book database.

Table: call

	ROWID	address	date	duration	flags	country_code	network_code
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	17015527494	1381853128	0	65540	310	260
2	2	17037876601	1382137390	6	4	310	260
3	3	15714855151	1382192016	67	4	310	260
4	4	17037876601	1382290201	0	5	310	260
5	5	15714855151	1382290246	23	5	310	260
6	6	15714855151	1382290323	6	5	310	260
7	7	17038990748	1382718325	0	65540	310	260
8	8	19546394558	1382735286	0	4	310	260
9	9	19546394558	1382887931	0	4	310	260
10	10	15714855151	1382999689	820	5	310	260
11	11	13013288997	1383581661	0	65540	310	260
12	12	+15714855151	1383788552	0	21	310	260
13	13	15714855151	1384088366	0	65540	310	260
14	14	+15714855151	1387129870	0	21	310	260
15	15	ebelford1@gmail.com	1387135183	775	21	310	260



The `ZCALLRECORD` in the `CallHistory.storedata` SQLite database contains a record for each call sent or received including FaceTime calls, similar to the `call_history.db`. iOS 8 device will also have this database.

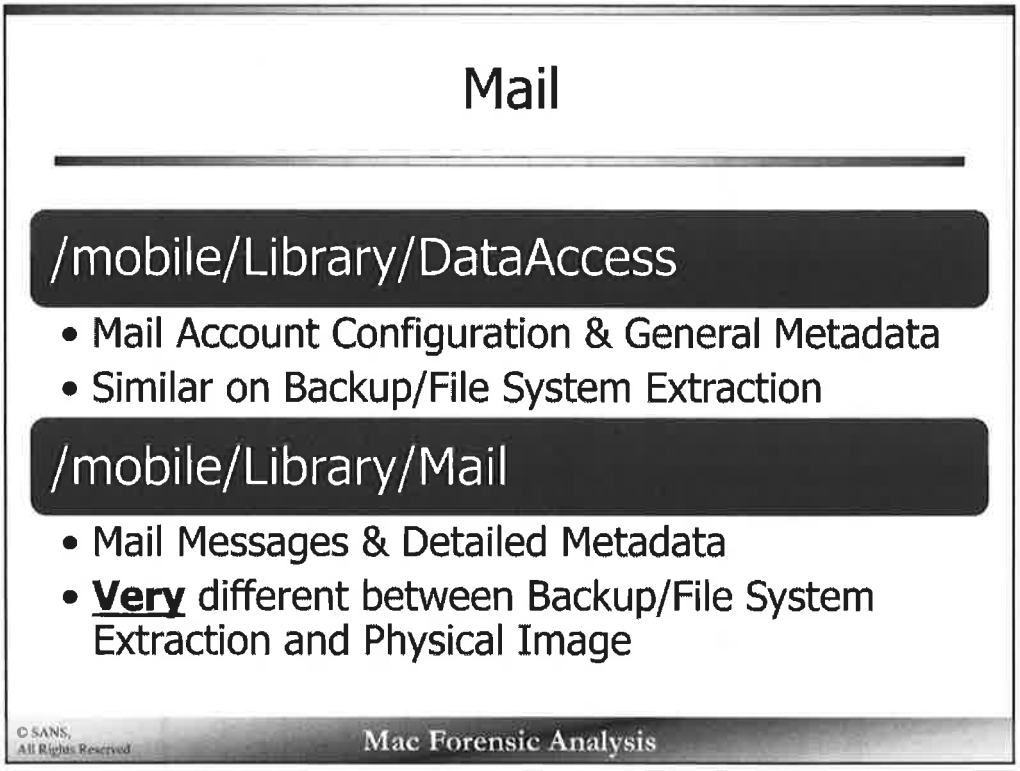
Each record contains the following data:

- `ZANSWERED` – 0 if outgoing call, 1 if incoming call
- `ZCALLTYPE`: (If duration is 0, call was missed/canceled)
  - 1 - Call
  - 8 – FaceTime
  - 16 – FaceTime Voice Call
- `ZFACE_TIME_DATA` – Data transferred (possibly in bytes) of FaceTime Call – More research is needed.
- `ZDURATION` – Time in seconds of the call.
- `ZADDRESS` – Contact phone number or e-mail address.
- `ZISO_COUNTRY_CODE` – Country code of contact.

Correlate the address with the data found in the Address Book database.

Table: ZCALLRECORD

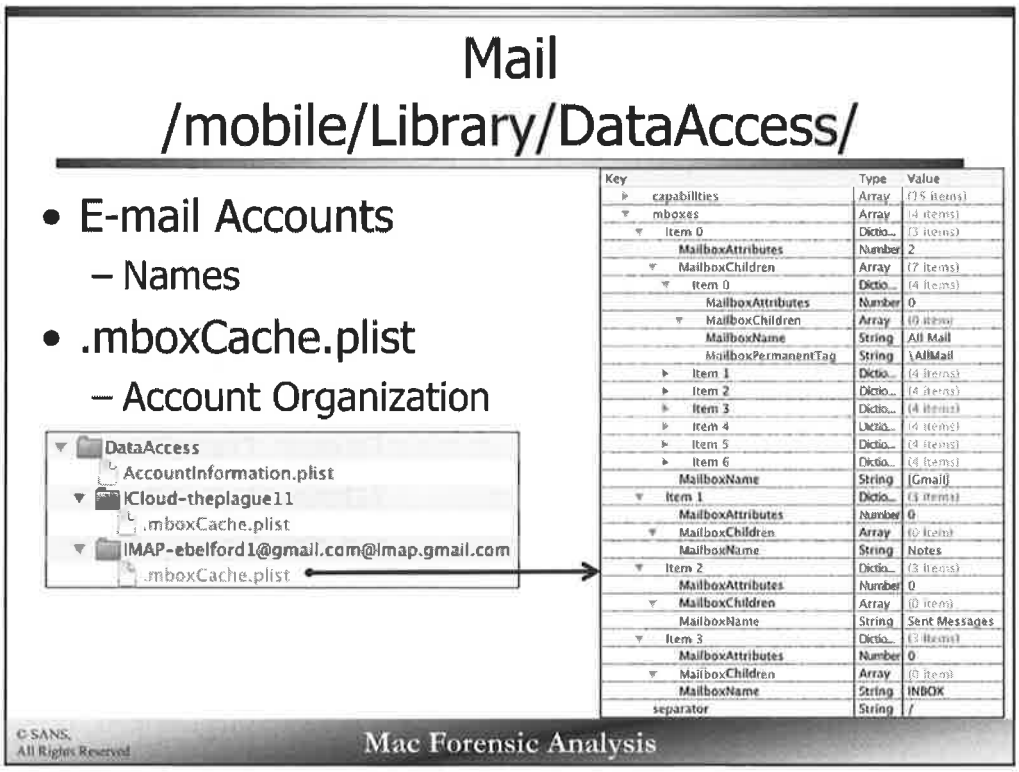
	Z_PK	ZANSWERED	ZCALLTYPE	ZFACE_TIME_DATA	ZDATE	ZDURATION	ZADDRESS	ZISO_COUNTRY_CODE
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
104	104	0	1		429262178	0.0	7737[redacted]	us
105	105	0	1		432166913	134.0	703[redacted]	us
106	106	0	8	5367304	405956174	34.0	7034[redacted]	us
107	107	1	1		424214764	1235.0	202[redacted]	us
108	108	0	8	47904003	427581548	201.0	+177[redacted]	dk
109	109	0	8	0	427579062	0.0	+177[redacted]	dk
110	110	0	1		426785370	0.0	5134[redacted]	us
111	111	0	8	3138434	421292801	84.0	7034[redacted]	us



The contents of e-mail is very different depending on what type of acquisition you have. Each acquisition type has two areas where e-mail data is found.

The `DataAccess` directory in /mobile/Library/ contains general e-mail metadata and e-mail account configuration information. This information is the same for all extractions.

The Mail directory in `/mobile/Library` is where things can be very different, depending on the type of acquisition you may have great information or you may have little to none.

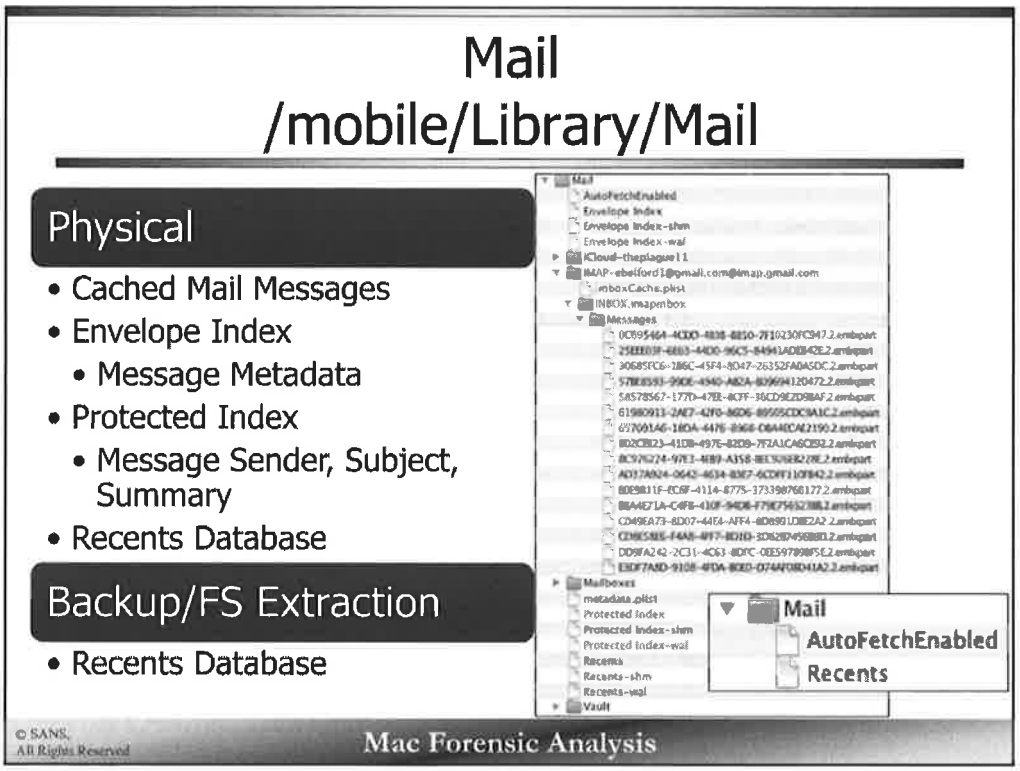


The `DataAccess` directory contains directories named after the e-mail accounts they represent. For example the iCloud-thePlague11 is an iCloud e-mail account (`theplague11@icloud.com`) while the other is a Gmail account (`ebelford1@gmail.com`).

The `.mboxCache.plist` files contain the organizational structure of each e-mail account. This property list file contains the folder names the e-mail messages are organized into.

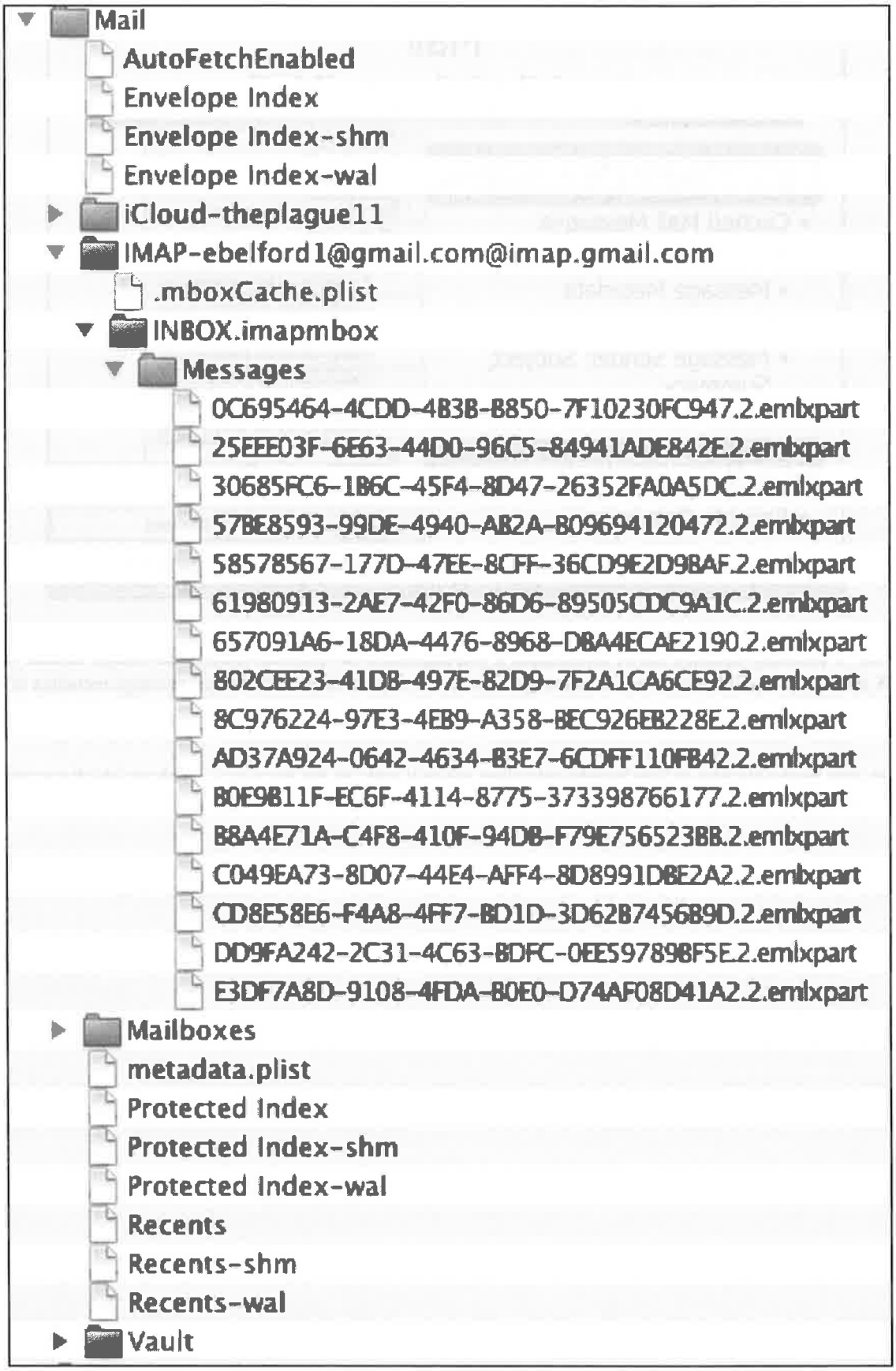
In these files you may find folder of interest depending on how the user has organized their e-mail.

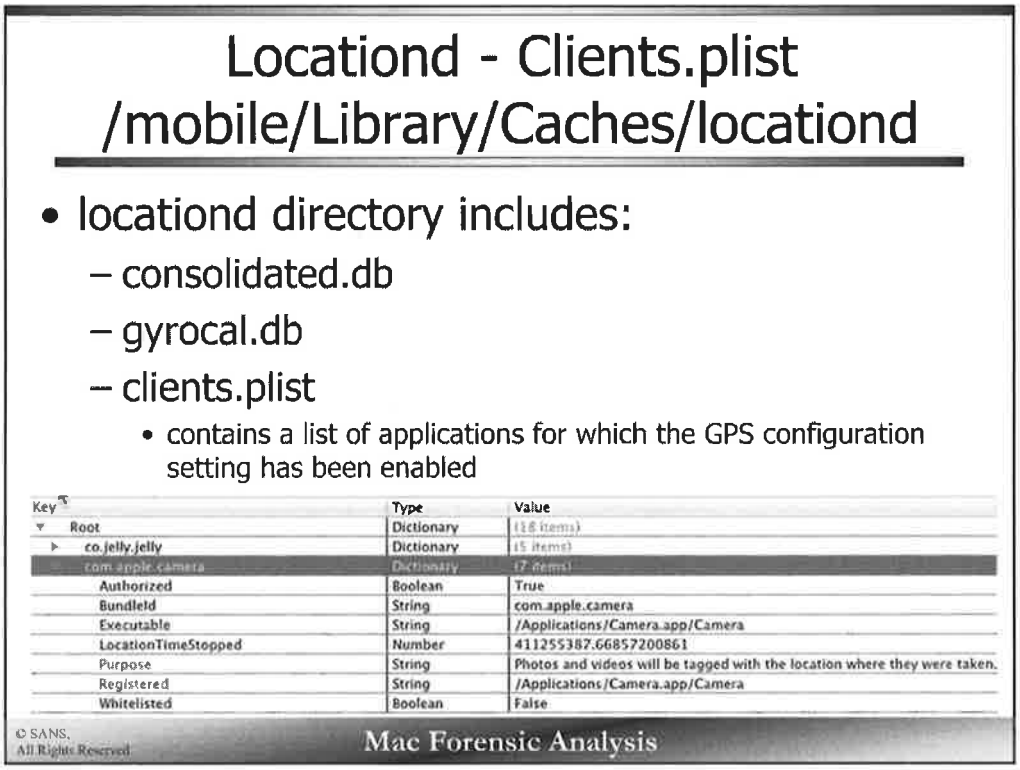
Key	Type	Value
capabilities	Array	(15 items)
mboxes	Array	(4 items)
Item 0	Dictio...	(3 items)
MailboxAttributes	Number	2
MailboxChildren	Array	(7 items)
Item 0	Dictio...	(4 items)
MailboxAttributes	Number	0
MailboxChildren	Array	(0 item)
MailboxName	String	All Mail
MailboxPermanentTag	String	\AllMail
Item 1	Dictio...	(4 items)
Item 2	Dictio...	(4 items)
Item 3	Dictio...	(4 items)
Item 4	Dictio...	(4 items)
Item 5	Dictio...	(4 items)
Item 6	Dictio...	(4 items)
MailboxName	String	[Gmail]
Item 1	Dictio...	(3 items)
MailboxAttributes	Number	0
MailboxChildren	Array	(0 item)
MailboxName	String	Notes
Item 2	Dictio...	(3 items)
MailboxAttributes	Number	0
MailboxChildren	Array	(0 item)
MailboxName	String	Sent Messages
Item 3	Dictio...	(3 items)
MailboxAttributes	Number	0
MailboxChildren	Array	(0 item)
MailboxName	String	INBOX
separator	String	/



The `Mail` directory contains lots of information if you acquired a full physical (decrypted) image. Similar to an OS X system, you will have full e-mail messages, the `Recents` database and detailed message metadata in the `Index` databases.

If you only have a Backup or File System extraction you will only get the `Recents` database which contains information about recently messaged contacts, similar to the same database on OS X.


Mail
AutoFetchEnabled
Envelope Index
Envelope Index-shm
Envelope Index-wal
iCloud-theplague11
IMAP-ebelford1@gmail.com@imap.gmail.com
.mboxCache.plist
INBOX.imapmbox
Messages
0C695464-4CDD-4B3B-B850-7F10230FC947.2.emlxpart
25EEE03F-6E63-44D0-96C5-84941ADE842E.2.emlxpart
30685FC6-1B6C-45F4-8D47-26352FA0A5DC.2.emlxpart
57BE8593-99DE-4940-AB2A-B09694120472.2.emlxpart
58578567-177D-47EE-8CFF-36CD9E2D9BAF.2.emlxpart
61980913-2AE7-42F0-86D6-89505CDC9A1C.2.emlxpart
657091A6-18DA-4476-8968-D8A4ECAE2190.2.emlxpart
802CEE23-41D8-497E-82D9-7F2A1CA6CE92.2.emlxpart
8C976224-97E3-4EB9-A358-BEC926EB228E.2.emlxpart
AD37A924-0642-4634-B3E7-6CDFF110FB42.2.emlxpart
B0E9B11F-EC6F-4114-8775-373398766177.2.emlxpart
B8A4E71A-C4F8-410F-94DB-F79E756523BB.2.emlxpart
C049EA73-8D07-44E4-AFF4-8D8991DBE2A2.2.emlxpart
CD8E58E6-F4A8-4FF7-BD1D-3D62B7456B9D.2.emlxpart
DD9FA242-2C31-4C63-BDFC-0EE597898F5E.2.emlxpart
E3DF7A8D-9108-4FDA-B0E0-D74AF08D41A2.2.emlxpart
Mailboxes
metadata.plist
Protected Index
Protected Index-shm
Protected Index-wal
Recents
Recents-shm
Recents-wal
Vault

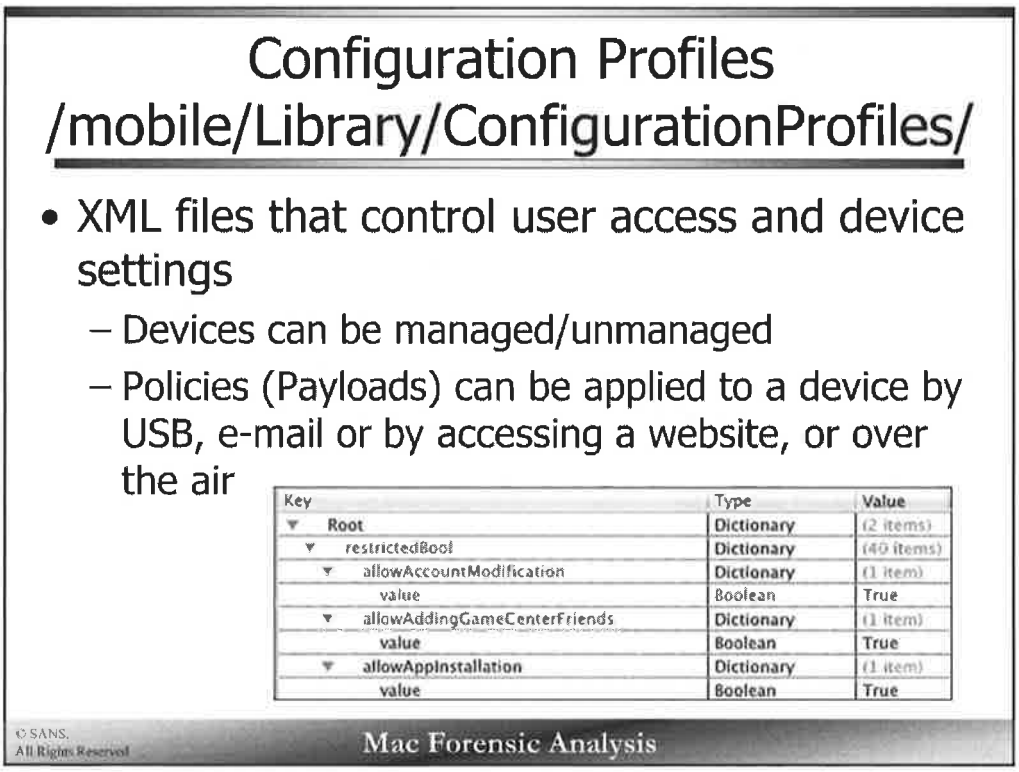


The locationd directory contains several files related to location information stored on an iOS device. The most notable may be the consolidated.db file, which was the cause for major concern when its contents were made public to many unsuspecting iOS users. This file contained wireless hotspots and cell towers locations within a certain radius of the device which was said to better allow the device to calculate its location rather than relying solely on GPS satellite data. This file once contained up to a year's worth of location data, but has since been revised in subsequent iOS firmware releases.

The most valuable file in the locationd directory now is the clients.plist file. This file maintains a list of all of the applications that have been granted GPS permissions. This is important because it may clue the examiner to pay extra detail to those applications listed in the file, as GPS coordinates are most likely stored along with other relevant user information.

Source [1] http://www.apple.com/pr/library/2011/04/27Apple-Q-A-on-Location-Data.html

Key	Type	Value
▼ Root	Dictionary	(18 items)
▶ co.jelly.jelly	Dictionary	(5 items)
com.apple.camera	Dictionary	(7 items)
Authorized	Boolean	True
BundleId	String	com.apple.camera
Executable	String	/Applications/Camera.app/Camera
LocationTimeStopped	Number	411255387.66857200861
Purpose	String	Photos and videos will be tagged with the location where they were taken.
Registered	String	/Applications/Camera.app/Camera
Whitelisted	Boolean	False



The directory, ConfigurationProfiles, contains files that relate to the management and polices/restrictions of the iOS device.

Device policies can be applied by direct connection to the device (USB), e-mail, sending users to a website or over-the-air. Reviewing the information in the ConfigurationProfiles directory will give insight into whether or not the device is managed by a Mobile Device Management (MDM) utility. MDMs can enforce policies related to Wi-Fi access, VPN, E-mail configuration and more.

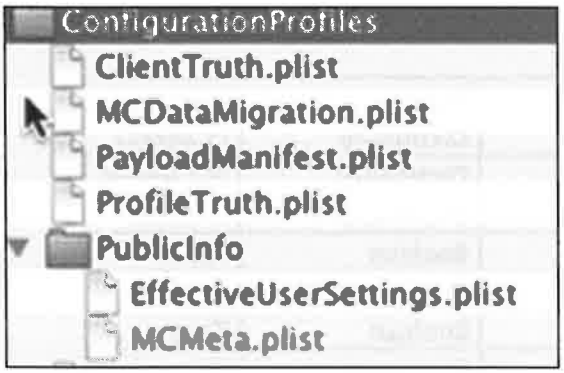
If a device is unmanaged no "Profiles" will be listed when navigating to Main Menu > Settings > General of the device. If the device is unmanaged, the important user settings can be reviewed by accessing the EffectiveUserSettings.plist file. These same settings can be accessed on the device by going to Settings > General > Restrictions. On unmanaged devices, these restrictions can be modified at will by the user.

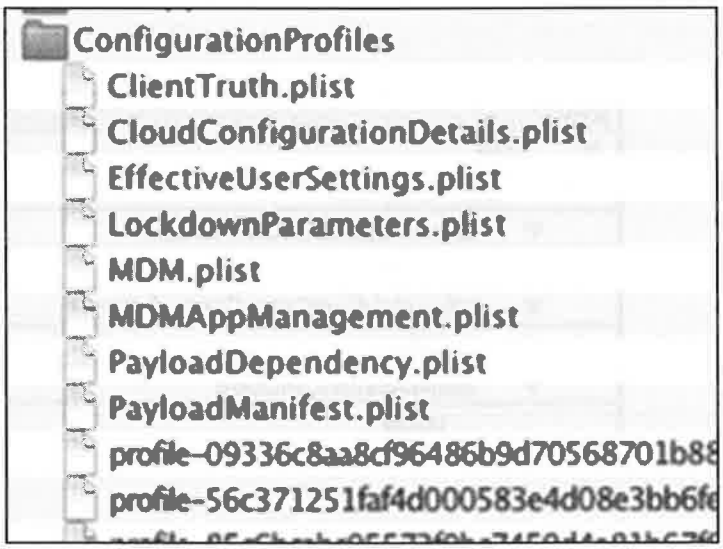
Key	Type	Value
▼ Root	Dictionary	(2 items)
▼ restrictedBool	Dictionary	(40 items)
▼ allowAccountModification	Dictionary	(1 item)
value	Boolean	True
▼ allowAddingGameCenterFriends	Dictionary	(1 item)
value	Boolean	True
▼ allowAppInstallation	Dictionary	(1 item)
value	Boolean	True

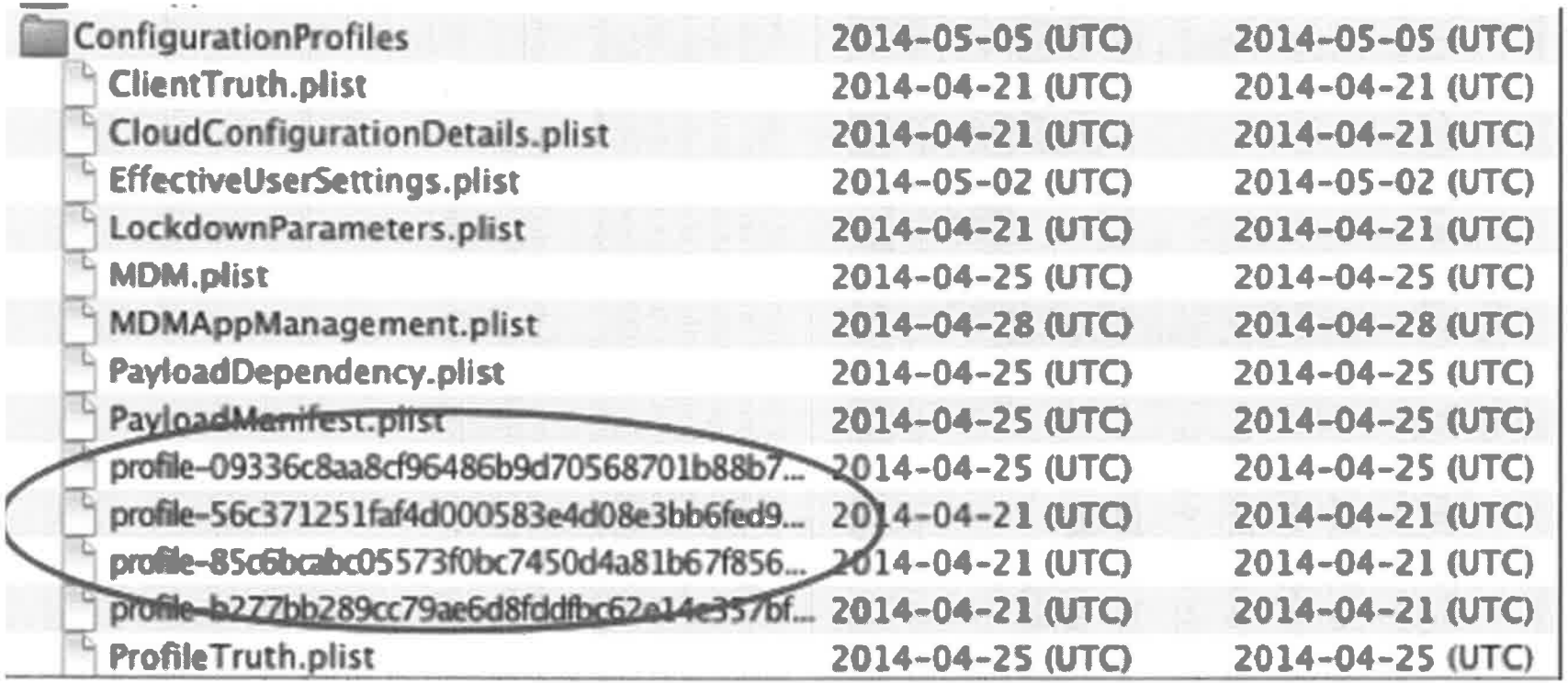
Managed devices will still contain an EffectiveUserSettings.plist file, but locating additional files in the ConfigurationProfile directory will confirm the existence of an MDM. The policies that have been applied by the MDM will trump those that are applied by the user under Restrictions from the device menu. There are many MDM solutions available for iOS devices and each will deposit different artifacts. Not all MDM solutions are created equally and this directory is a good starting point for forensic recovery of applications controlled by an MDM.

Reviewing files with the .stub extension can give insight into which organization may be deploying the MDM solution as well as how that MDM profile was pushed to the device.

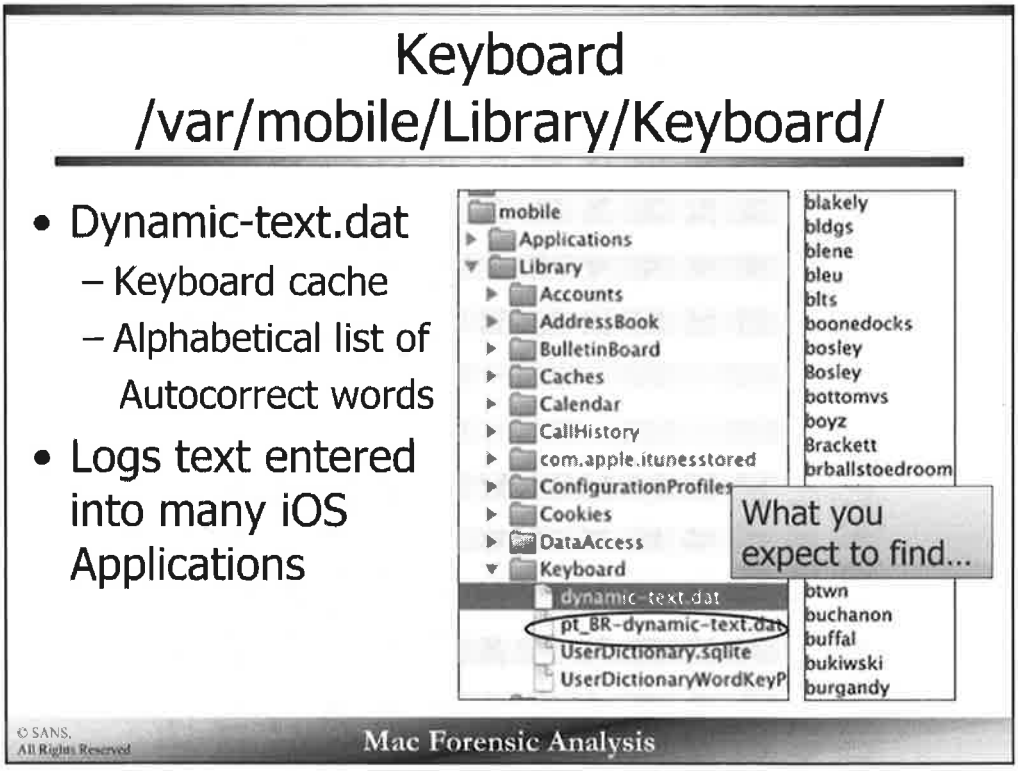
Source [1] https://www.apple.com/iphone/business/it/management.html







ConfigurationProfiles	2014-05-05 (UTC)	2014-05-05 (UTC)
ClientTruth.plist	2014-04-21 (UTC)	2014-04-21 (UTC)
CloudConfigurationDetails.plist	2014-04-21 (UTC)	2014-04-21 (UTC)
EffectiveUserSettings.plist	2014-05-02 (UTC)	2014-05-02 (UTC)
LockdownParameters.plist	2014-04-21 (UTC)	2014-04-21 (UTC)
MDM.plist	2014-04-25 (UTC)	2014-04-25 (UTC)
MDMAppManagement.plist	2014-04-28 (UTC)	2014-04-28 (UTC)
PayloadDependency.plist	2014-04-25 (UTC)	2014-04-25 (UTC)
PayloadManifest.plist	2014-04-25 (UTC)	2014-04-25 (UTC)
profile-09336c8aa8cf96486b9d70568701b88b7...	2014-04-25 (UTC)	2014-04-25 (UTC)
profile-56c371251faf4d000583e4d08e3bb6fed9...	2014-04-21 (UTC)	2014-04-21 (UTC)
profile-85c6bcabc05573f0bc7450d4a81b67f856...	2014-04-21 (UTC)	2014-04-21 (UTC)
profile-b277bb289cc79ae6d8fddfbc62e14e357bf...	2014-04-21 (UTC)	2014-04-21 (UTC)
ProfileTruth.plist	2014-04-25 (UTC)	2014-04-25 (UTC)

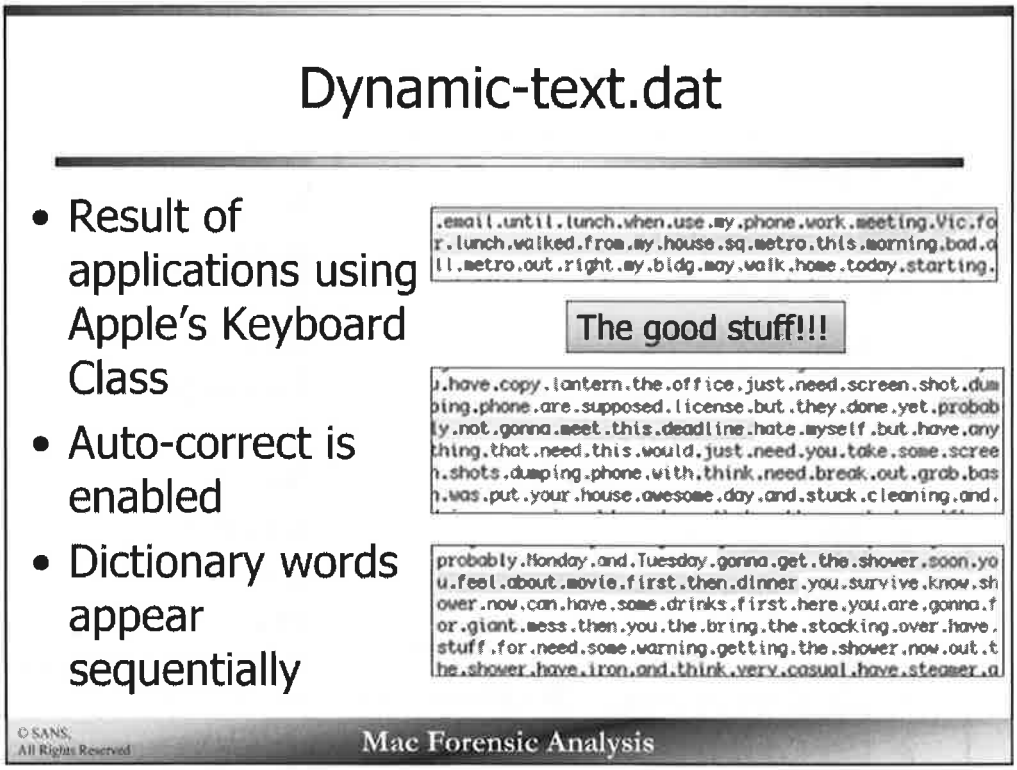


The file, dynamic-text.dat functions as part of the auto-correct feature for iOS devices. Based on the content that can be recovered from the file, it has also been called "Apple's keylogger." This file can contain some very important user input that may not otherwise be captured during a file system or logical acquisition of the device. This file typically contains words, and non words alike, in the form of an alphabetical list which may not yield any real forensic value. This dat file can be opened with a Hex editor or any other text application. The screen shot above was viewed as "strings" from within the BlackLight application. Alternatively, the file can be exported from a free tool like iBackupBot and opened in a Text Editor or any other word-processing application.

Do not overlook dynamic-text.dat files like the one circled above, which will contain input data in various languages as entered on the device (eg. Pt_BR-dynamic-text.dat = Brazil/Portuguese).

Source: [1] http://www.chmag.in/article/aug2012/apple-ios-vulnerabilities


mobile
Applications
Library
Accounts
AddressBook
BulletinBoard
Caches
Calendar
CallHistory
com.apple.itunesstored
ConfigurationProfiles
Cookies
DataAccess
Keyboard
dynamic-text.dat
pt_BR-dynamic-text.dat
UserDictionary.sqlite
UserDictionaryWordKeyP
blakely
bldgs
blene
bleu
blts
boonedocks
bosley
Bosley
bottomvs
boyz
Brackett
brballstoedroom
What you expect to find...
btwn
buchanon
buffal
bukiwski
burgandy



Of more interest than the alphabetical list of terms, is the cache of phrases often in sequential order just as the data was typed into an application. Most applications make use of Apple's native Keyboard Class, so this data will be retained by default. Often times, E-mail messages, Facebook and Twitter posts, and chats can be recovered almost in their entirety from this file. This file often lacks other important contextual data due to the storage format (strings), but a keyword search targeting this file can often return positive leads for further analysis.

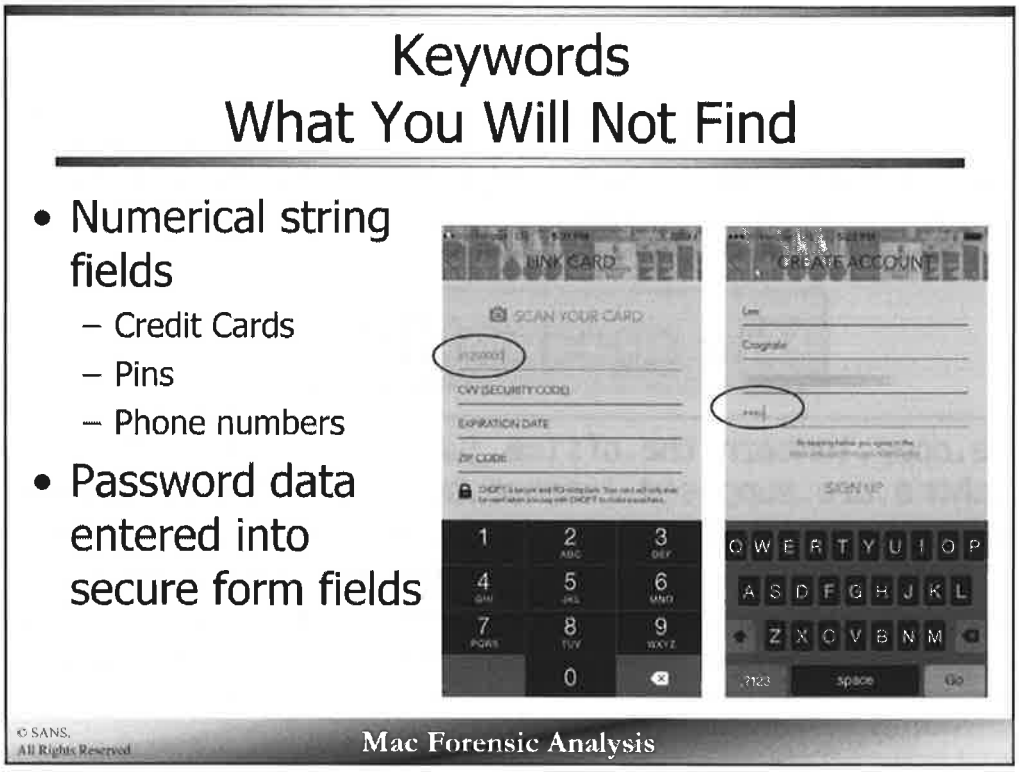
Source: [1]
https://developer.apple.com/library/ios/documentation/ToolsLanguages/Reference/UIAKeyboardClassReference/UIAKeyboard/UIAKeyboard.html – Applie's Keyboard Class

.email.until.lunch.when.use.my.phone.work.meeting.Vic.fo
r.lunch.walked.from.my.house.sq.metro.this.morning.bad.a
ll.metro.out.right.my.bldg.may.walk.home.today.starting.

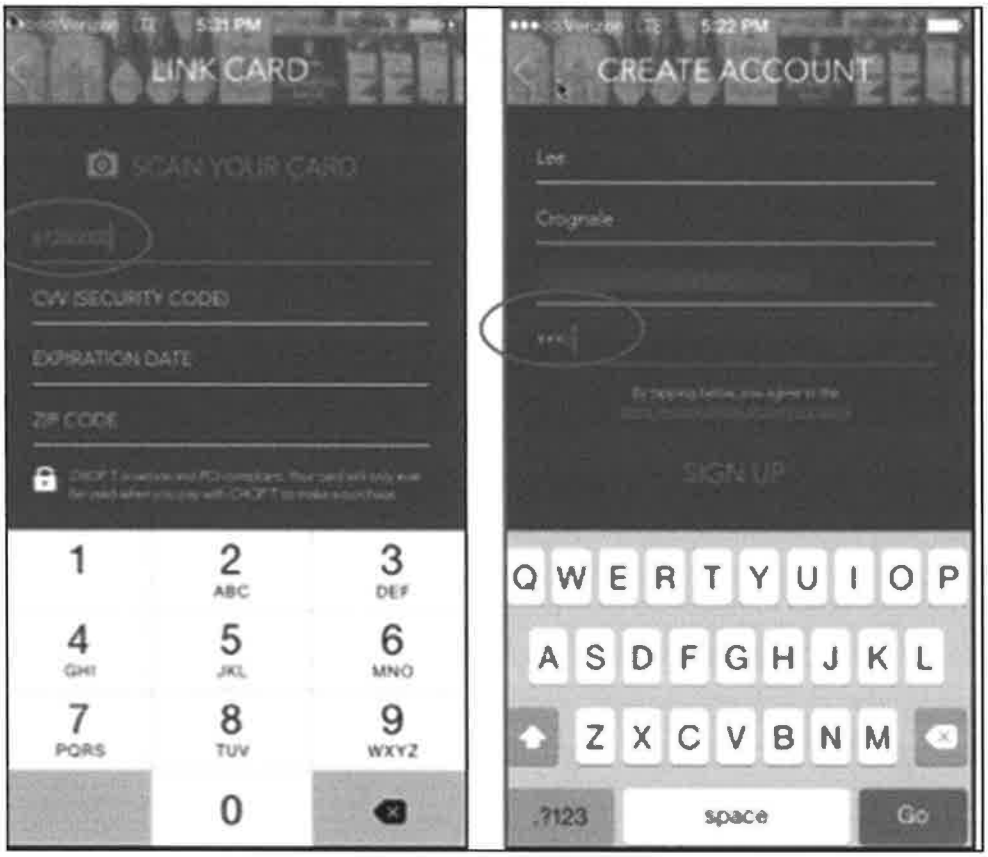
The good stuff!!!

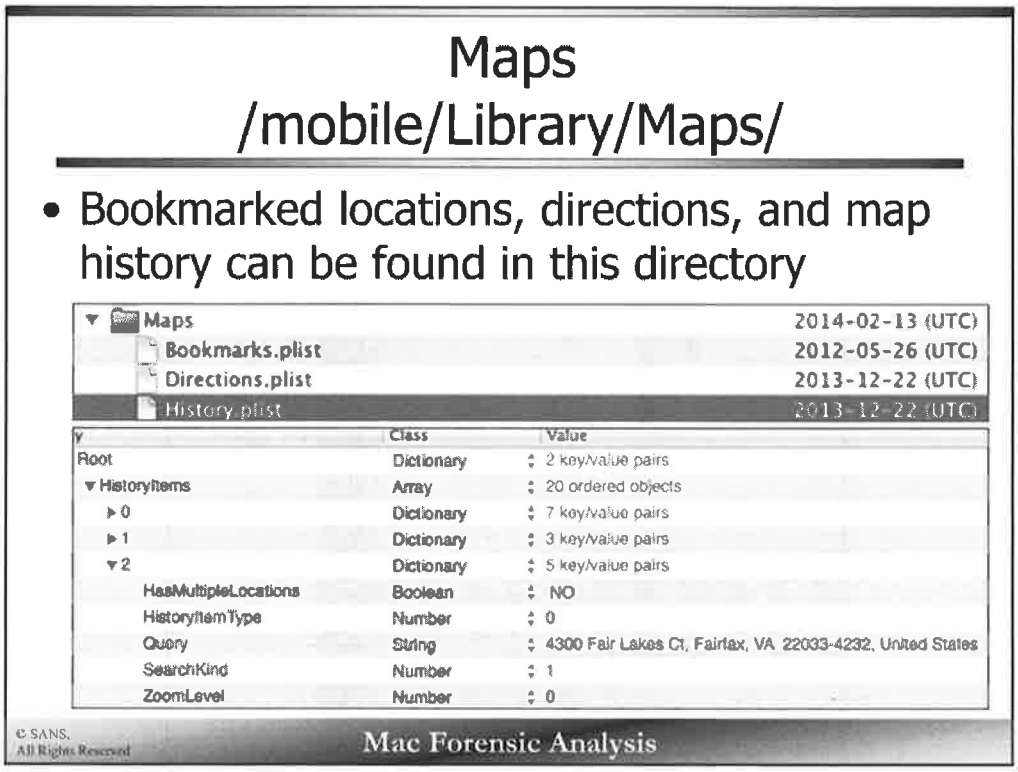
u.have.copy.lantern.the.office.just.need.screen.shot.dum
bing.phone.are.supposed.license.but.they.done.yet.probab
ly.not.gonna.meet.this.deadline.hate.myself.but.have.any
thing.that.need.this.would.just.need.you.take.some.scree
n.shots.dumping.phone.with.think.need.break.out.grab.bas
n.was.put.your.house.awesome.day.and.stuck.cleaning.and.

probably.Monday.and.Tuesday.gonna.get.the.shower.soon.yo
u.feel.about.movie.first.then.dinner.you.survive.know.sh
ower.now.can.have.some.drinks.first.here.you.are.gonna.f
or.giant.mess.then.you.the.bring.the.stocking.over.have.
stuff.for.need.some.warning.getting.the.shower.now.out.t
he.shower.have.iron.and.think.very.casual.have.steamer.a



This DOES not apply to usernames and passwords that are entered into other unsecure applications like Notes, Calendars, SMS and E-mail. Applications that do not use secure form fields could be vulnerable.



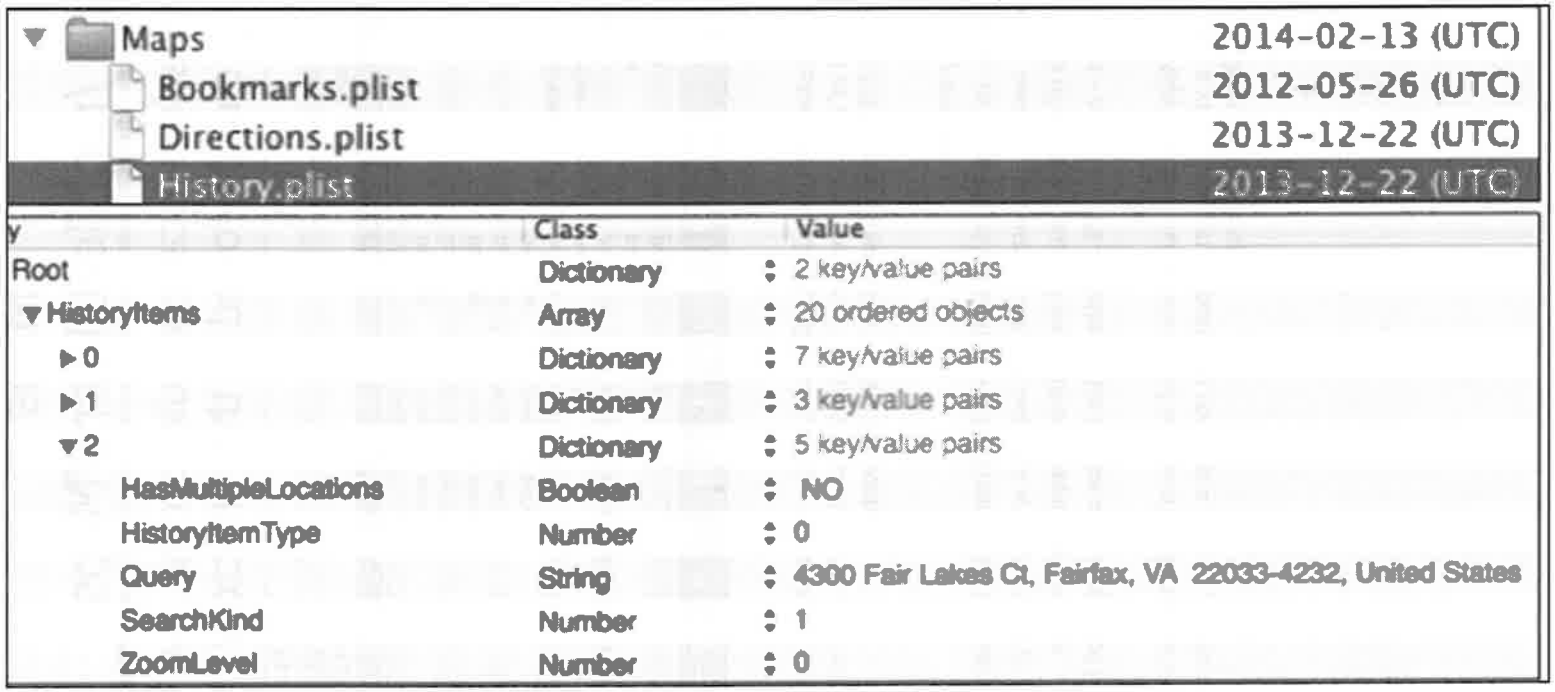


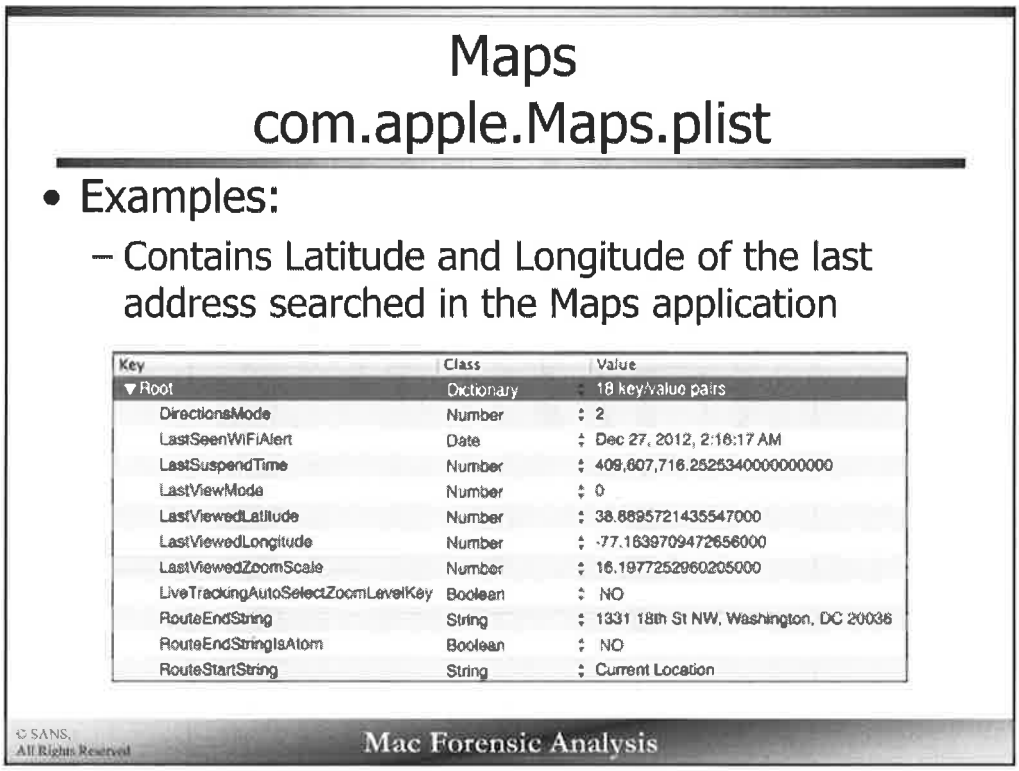
The mobile/Library/Maps directory will contain plist files with content pertaining to data saved by the iOS built-in mapping application. Most of the data is contained in binary plist files, but some require opening the file in something other than the Mac OS Property List Editor to view the content.

The maps folder will typically contain:

- Bookmarks.plist – contains location data that was bookmarked by the user
- Directions.plist – contains the address that was searched and possible web-content
- History.plist – list of addresses entered into built-in Mapping application
- SearchResults.dat – contains GPS coordinates for the last address entered in the Map Search field

Content recovered from the Maps directory can be encrypted so exporting the file for viewing within native Mac OS applications is not always a viable solution. BlackLight contains a built-in SQLite viewer and the data strings can also be reviewed from within the tool.


Maps
2014-02-13 (UTC)
Bookmarks.plist
2012-05-26 (UTC)
Directions.plist
2013-12-22 (UTC)
History.plist
2013-12-22 (UTC)
Class
Value
Root
Dictionary
2 key/value pairs
HistoryItems
Array
20 ordered objects
0
Dictionary
7 key/value pairs
1
Dictionary
3 key/value pairs
2
Dictionary
5 key/value pairs
HasMultipleLocations
Boolean
NO
HistoryItemType
Number
0
Query
String
4300 Fair Lakes Ct, Fairfax, VA 22033-4232, United States
SearchKind
Number
1
ZoomLevel
Number
0



The maps.plist file contains information related to the Maps application. When an address is searched from within the Mapping application, the last GPS coordinates as well as the address will be contained in the file.

In addition to the plist files mentioned above under the mobile/Library/Preferences sub-directory, there are plist files relating to configuration settings located under mobile/SystemConfigurations and mobile/Library/ConfigurationProfiles.

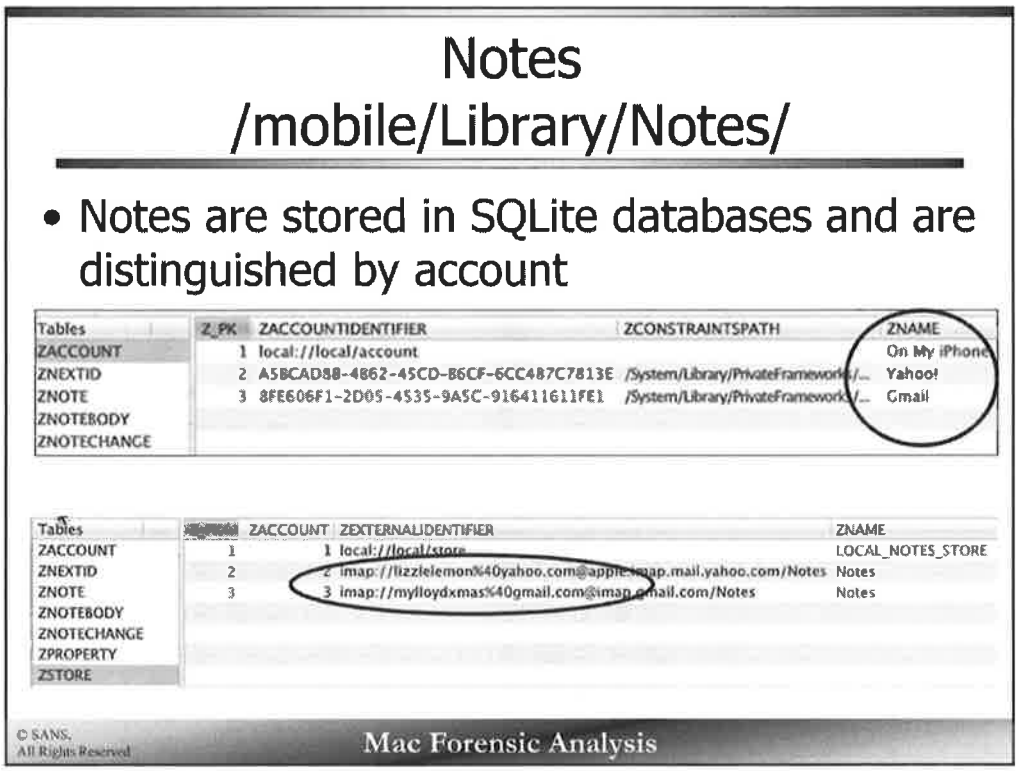
The files under SystemConfiguration include:

- SystemConfiguration/com.apple.accounts.exists.plist – accounts setup or disabled on the device
- SystemConfiguration/com.apple.AutoWake.plist – WIFI information
- SystemConfiguration/com.apple.mobilegestalt - device name
- SystemConfiguration/com.apple.network.identification.plist - Network information
- com.apple.radios.plist - Airplane Mode toggle
- SystemConfiguration/com.apple.wifi.plist - Network information
- SystemConfiguration/preferences.plist - Network information

Files under ConfigurationProfiles include preference settings related to Mobile Device Management (MDM) solutions applied to the device. Review these files for information on restricted applications/permissions and other application information related to specific MDM solutions.

A more comprehensive listing of the preference files and the data they can contain is located in the Appendix.

Key	Class	Value
▼ Root	Dictionary	18 key/value pairs
DirectionsMode	Number	2
LastSeenWiFiAlert	Date	Dec 27, 2012, 2:16:17 AM
LastSuspendTime	Number	409,607,716.2525340000000000
LastViewMode	Number	0
LastViewedLatitude	Number	38.8895721435547000
LastViewedLongitude	Number	-77.1639709472656000
LastViewedZoomScale	Number	16.1977252960205000
LiveTrackingAutoSelectZoomLevelKey	Boolean	NO
RouteEndString	String	1331 18th St NW, Washington, DC 20036
RouteEndStringIsAtom	Boolean	NO
RouteStartString	String	Current Location



Notes.sqlite Tables of importance

ZACCOUNT/ZSTORE – details the number of accounts syncing with the device

ZNOTE and ZNOTEBODY – will contain the message data

Tables

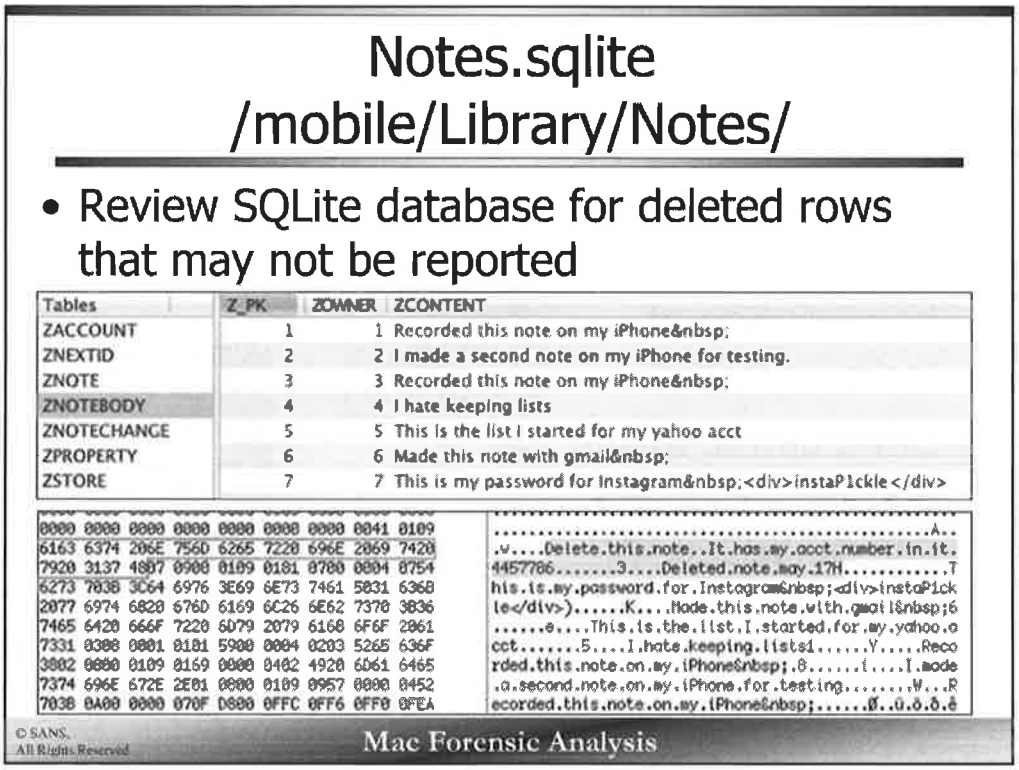
- ZACCOUNT
- ZNEXTID
- ZNOTE
- ZNOTEBODY
- ZNOTECHANGE

Z_PK	ZACCOUNTIDENTIFIER	ZCONSTRAINTSPATH	ZNAME
1	local://local/account		On My iPhone
2	A5BCAD8B-4B62-45CD-B6CF-6CC487C7813E	/System/Library/PrivateFrameworks/...	Yahoo!
3	8FE606F1-2D05-4535-9A5C-916411611FE1	/System/Library/PrivateFrameworks/...	Gmail

Tables

- ZACCOUNT
- ZNEXTID
- ZNOTE
- ZNOTEBODY
- ZNOTECHANGE
- ZPROPERTY
- ZSTORE

Z_PK	ZACCOUNT	ZEXTERNALIDENTIFIER	ZNAME
1	1	local://local/store	LOCAL_NOTES_STORE
2	2	imap://lizzlelemon%40yahoo.com@apple.imap.mail.yahoo.com/Notes	Notes
3	3	imap://mylloydxmas%40gmail.com@imap.gmail.com/Notes	Notes



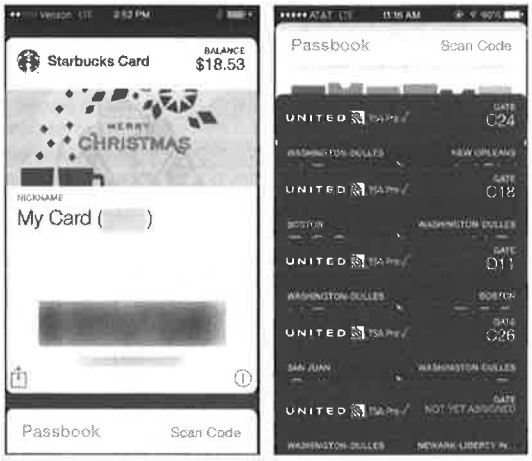
Depending on the analysis tool used for examination, deleted SQLite database entries may not be presented. Review the raw file for any deleted notes entries if you have confirmed that your tool does not support the recovery of data from SQLite databases.

Tables	Z_PK	ZOWNER	ZCONTENT
ZACCOUNT	1	1	Recorded this note on my iPhone
ZNEXTID	2	2	I made a second note on my iPhone for testing.
ZNOTE	3	3	Recorded this note on my iPhone
ZNOTEBODY	4	4	I hate keeping lists
ZNOTECHANGE	5	5	This is the list I started for my yahoo acct
ZPROPERTY	6	6	Made this note with gmail
ZSTORE	7	7	This is my password for Instagram instaP1ckle

```
0000 0000 0000 0000 0000 0000 0000 0041 0109  .....A..
6163 6374 206E 756D 6265 7220 696E 2069 7420  .w....Delete.this.note..It.has.my.acct.number.in.it.
7920 3137 4807 0900 0109 0181 0700 0004 0754  4457786.....3....Deleted.note.may.17H.....T
6273 703B 3C64 6976 3E69 6E73 7461 5031 636B  his.is.my.password.for.Instagram <div>instaP1ck
2077 6974 6820 676D 6169 6C26 6E62 7370 3B36  le</div>).....K....Made.this.note.with.gmail 6
7465 6420 666F 7220 6D79 2079 6168 6F6F 2061  .....e....This.is.the.list.I.started.for.my.yahoo.a
7331 0308 0001 0101 5900 0004 0203 5265 636F  cct.....5....I.hate.keeping.lists1.....Y.....Reco
3802 0800 0109 0169 0000 0402 4920 6D61 6465  rded.this.note.on.my.iPhone .8.....i....I.made
7374 696E 672E 2E01 0800 0109 0957 0000 0452  .a.second.note.on.my.iPhone.for.testing.....W...R
703B 0A00 0000 070F D800 0FFC 0FF6 0FF0 0FEA  ecorded.this.note.on.my.iPhone .....Ø..ü.ö.ð.ê
```

Passes
/mobile/Library/Passes/

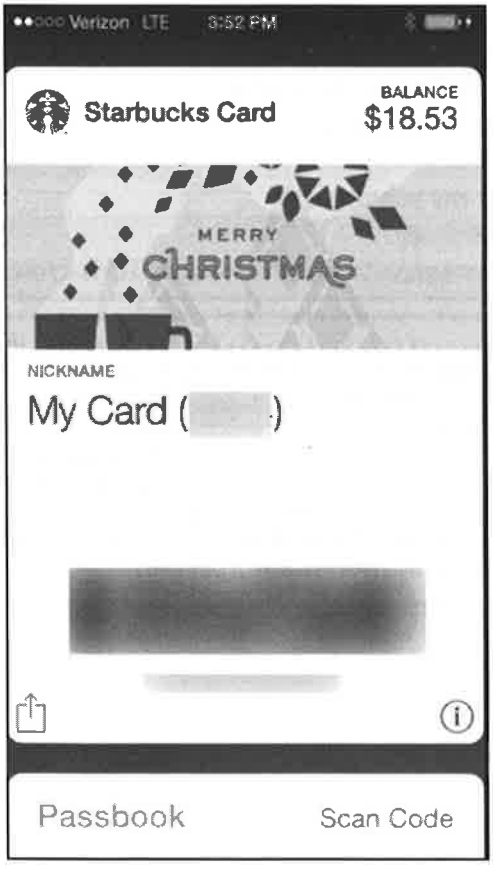
- Passbook data will appear on homescreen of locked devices
- Store cards, movie and airline tickets can be added for easy access

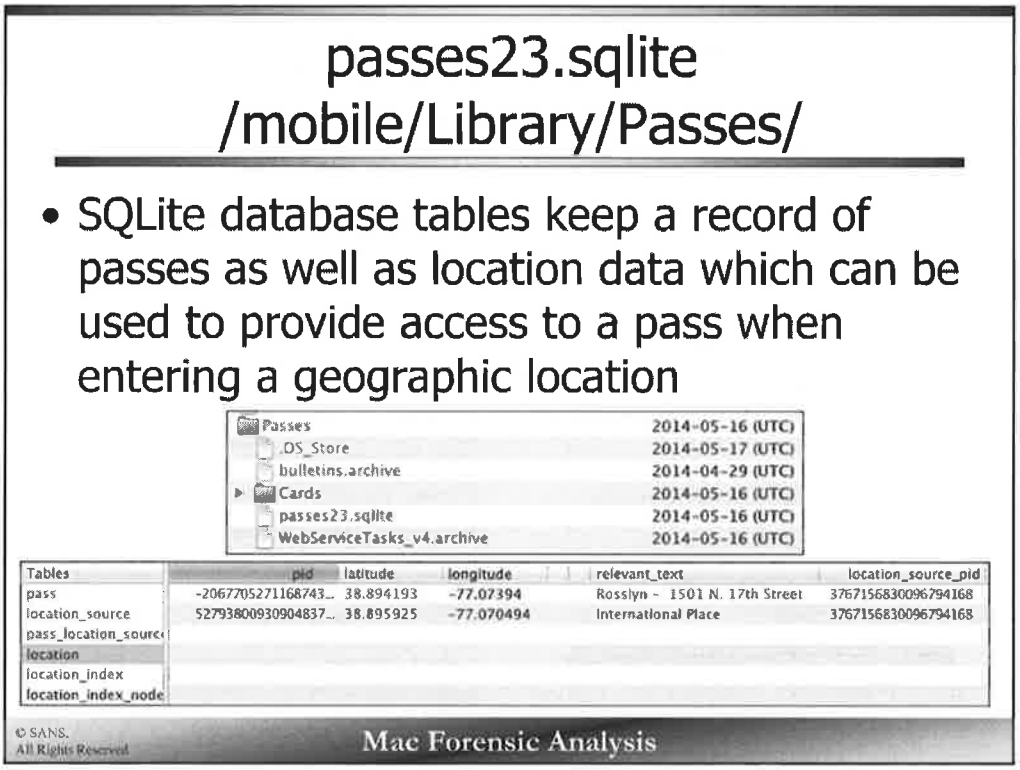


© SANS, All Rights Reserved

Pass data are movie tickets, airline tickets, restaurant and store cards that are managed in one location for ease of use. Passes can make use of geo-location settings as well as push notifications.

*The passbook images included above are screen captures of the device itself.





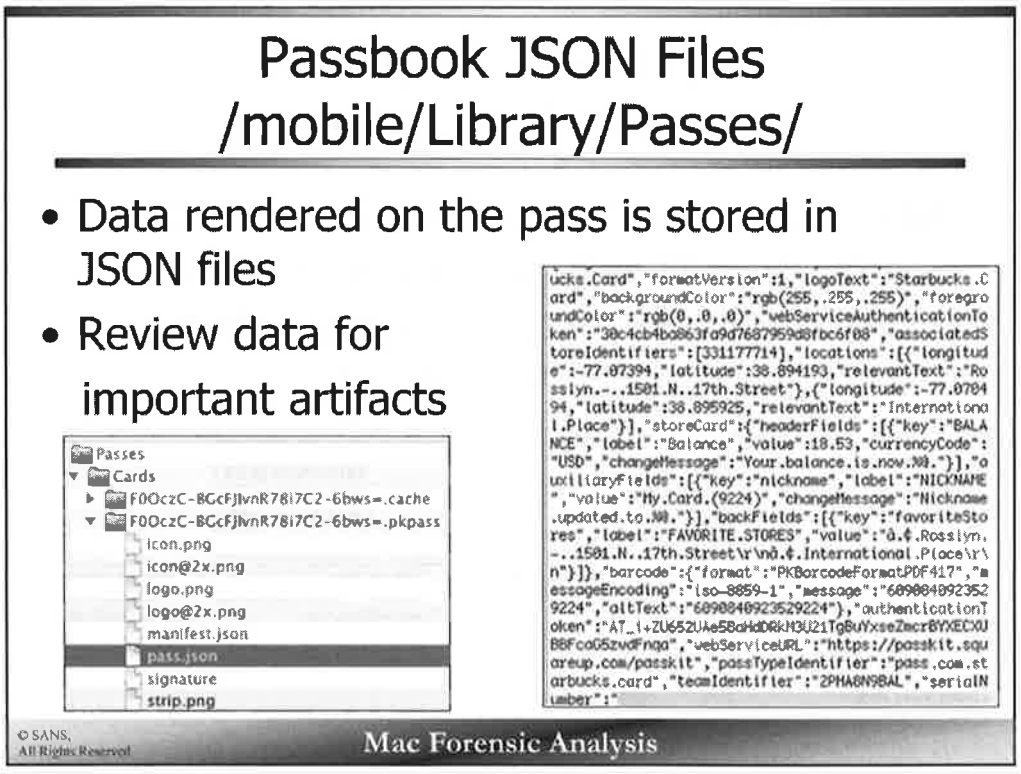
The passes23 database stores relevant information for each pass.

Passes are logged according to type (ex: Mobile Boarding pass/United or Starbucks). Location data can be added to each pass so that they appear on the home screen when you enter a pre-defined geographic area. Two Starbucks locations have been saved as favorite locations in the example above.

Within each Passes directory is a sub-directory which is used to store the various passes that have been entered by the user. Each pass will utilize two sub-folders to store information about the "pass". Files with the .archive file extension are plist files which can be opened in a standard viewer. The plists contain embedded PNG files which are used to store the logos that appear on each of the passes. The PNG files do not store the complete pass image which includes all of the user data for each pass.

Passes	2014-05-16 (UTC)
.DS_Store	2014-05-17 (UTC)
bulletins.archive	2014-04-29 (UTC)
▶ Cards	2014-05-16 (UTC)
passes23.sqlite	2014-05-16 (UTC)
WebServiceTasks_v4.archive	2014-05-16 (UTC)

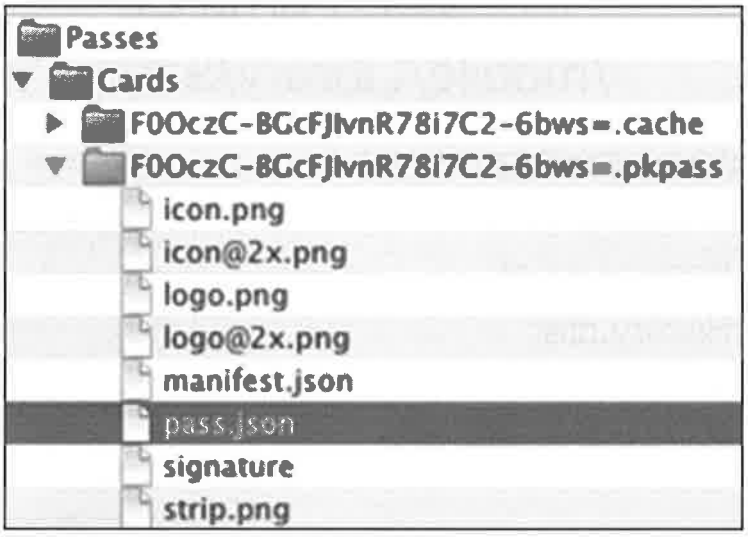
Tables	pid	latitude	longitude	relevant_text	location_source_pid
pass	-2067705271168743...	38.894193	-77.07394	Rosslyn - 1501 N. 17th Street	3767156830096794168
location_source	52793800930904837...	38.895925	-77.070494	International Place	3767156830096794168
pass_location_sourc					
location					
location_index					
location_index_node					



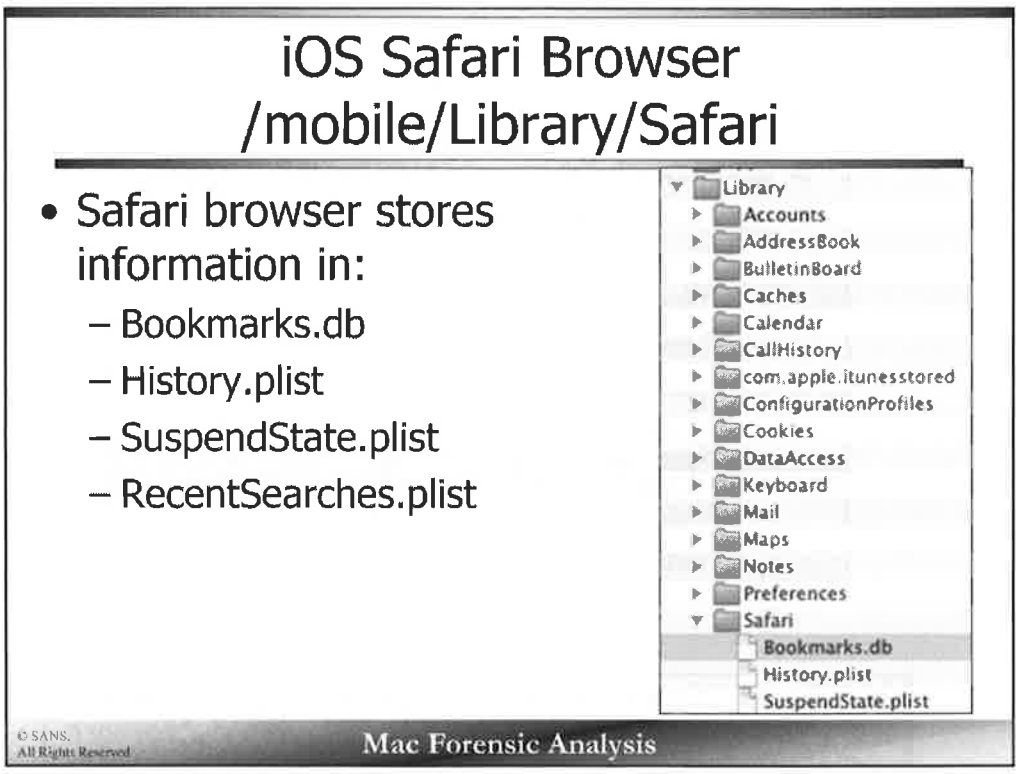
JSON files are used to store all of the content relevant for each "pass". Data is stored in the pass.json file in plain text and can be reviewed with any hex editing tools.

You can expect to find:

- Frequent flyer numbers
- Usernames
- Full First name and Last name (airline tickets)
- Flight Details
- Location information
- And many more interesting artifacts dependant on the type of passes that are available




```
ucks.Card","formatVersion":1,"logoText":"Starbucks.C
ard","backgroundColor":"rgb(255,.255,.255)","foregro
undColor":"rgb(0,.0,.0)","webServiceAuthenticationTo
ken":"30c4cb4ba863fa9d7687959d8fbc6f08","associatedS
toreIdentifiers":[331177714],"locations":[{"longitud
e":-77.07394,"latitude":38.894193,"relevantText":"Ro
sslyn.-..1501.N..17th.Street"},{"longitude":-77.0704
94,"latitude":38.895925,"relevantText":"Internationa
l.Place"}],"storeCard":{"headerFields":[{"key":"BALA
NCE","label":"Balance","value":18.53,"currencyCode":
"USD","changeMessage":"Your.balance.is.now.%@."}],"a
uxiliaryFields":[{"key":"nickname","label":"NICKNAME
","value":"My.Card.(9224)","changeMessage":"Nickname
.updated.to.%@."}],"backFields":[{"key":"favoriteSto
res","label":"FAVORITE.STORES","value":"â.¢.Rosslyn.
-..1501.N..17th.Street\r\nâ.¢.International.Place\r\
n"}]},"barcode":{"format":"PKBarcodeFormatPDF417","m
essageEncoding":"iso-8859-1","message":"6090840923529224","altText":"6090840923529224"},"authenticationT
oken":"AT_i+ZU652UAe5BaHdDRkM3U21TgBuYxseZwcrBYXECXU
B8FcaG5zwdFnqa","webServiceURL":"https://passkit.squ
areup.com/passkit","passTypeIdentifier":"pass.com.st
arbucks.card","teamIdentifier":"2PHA8N9BAL","serialN
umber":"
```

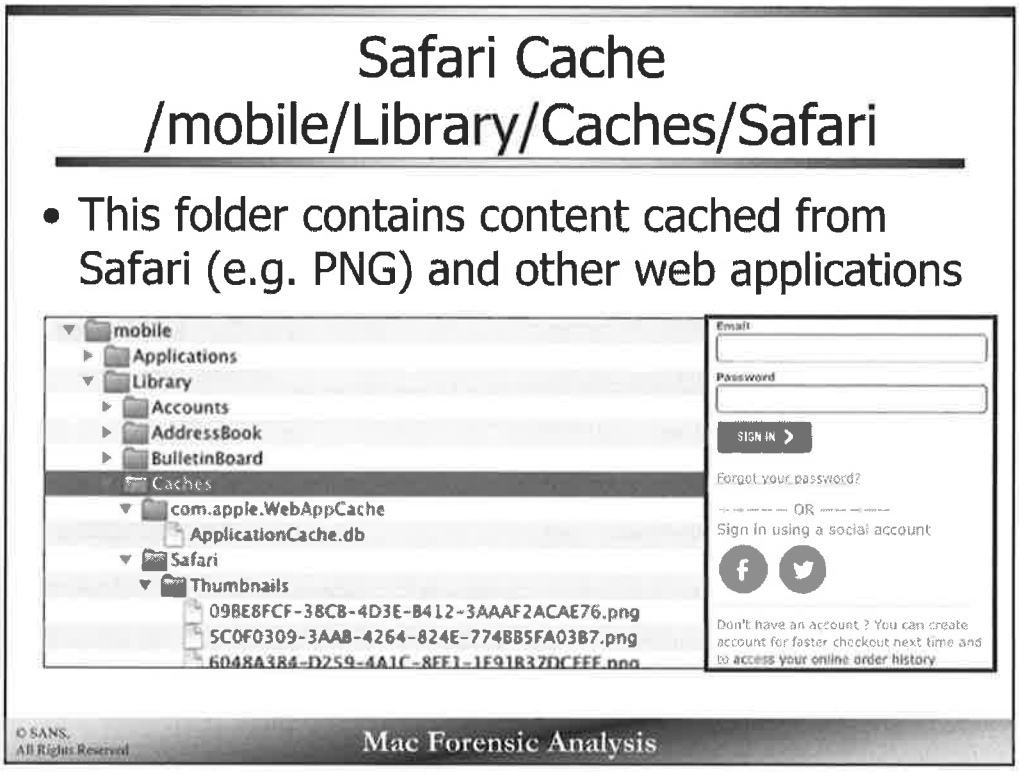


In earlier versions of iOS, the files of interest were stored under the Library/Safari directory

- Mobile/Library/Safari/Bookmarks.db – contains Safari bookmarked data
- Mobile/Library/Safari/History.plist – a cumulative list of browser history data since (resets after the browser cache is cleared)
- Mobile/Library/Safari/Suspendstate.plist – a glimpse of the last browser history before the Safari browser was exited. This could be due to an application crash, turning off the phone, or pressing the Home key to return to the main menu screen.This plist file will reflect the state of the browser the last time it was accessed.

In iOS 7.x, also review the mobile/Library/Safari directory as well as the mobile/applications/com.apple.mobilesafari/Library/Safari directory for all of the files related to the Safari browser.

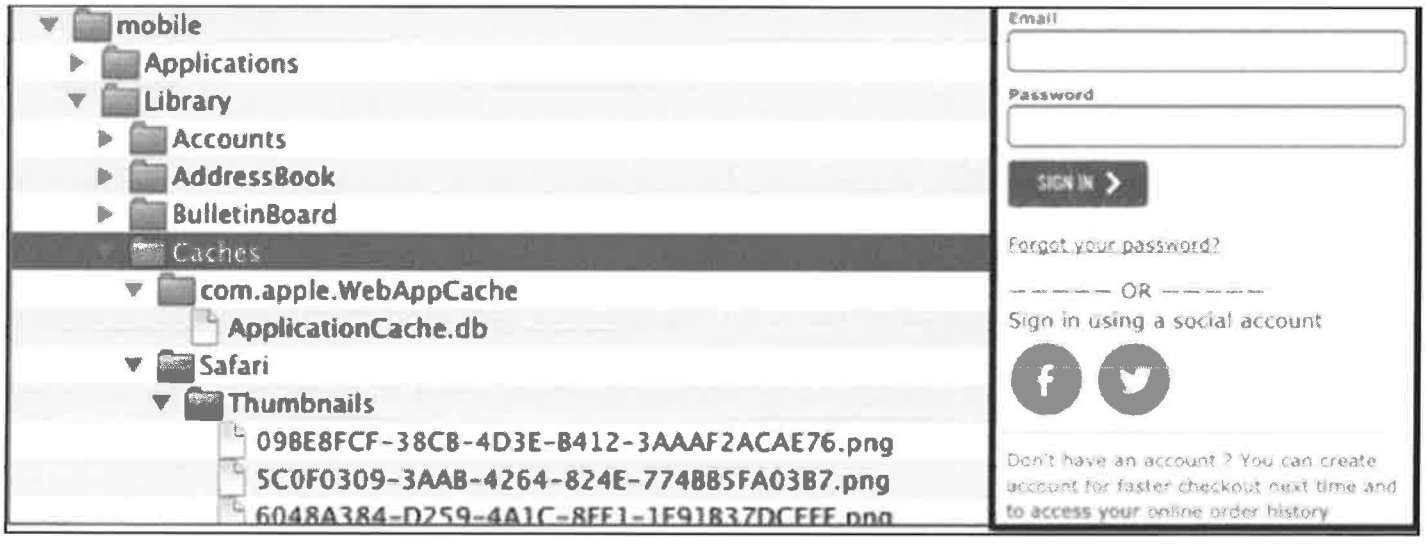
- mobile/Library/Safari/Bookmarks.db
- mobile/applications/com.apple.mobilesafari/Library/Safari/RecentSearches
- mobile/applications/com.apple.mobilesafari/Library/Safari/History.plist
- mobile/applications/com.apple.mobilesafari/Library/Safari/SuspendState.plist

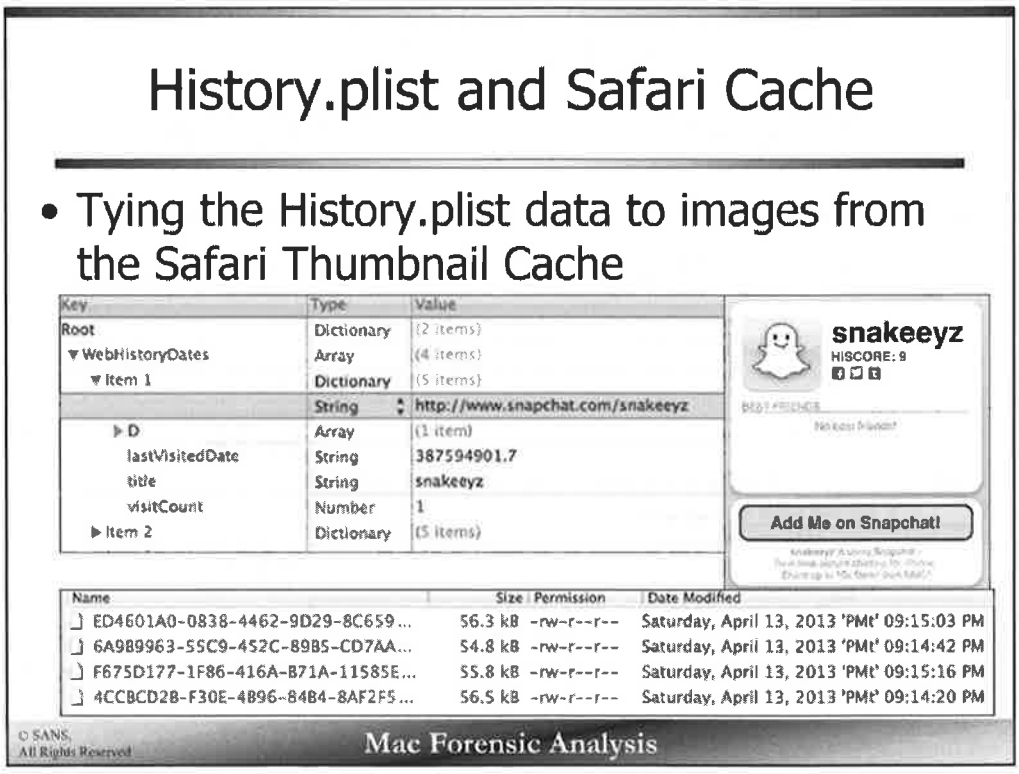


The Caches directory contains multiple items of forensic importance.

There may be several sub-directories in this location to include: com.apple.WebAppCache, Safari, and locationd.

com.apple.WebAppCache/ApplicationCache is a database file containing data that is accessed when accessing an application through the Safari Browser and not the native application interface. The Safari/Thumbnails directory contains a listing of images viewed in the Safari Browser. Web browsing images are stored as PNG files.





The Safari directory and the thumbnail directory should be reviewed together. Thumbnails of recently viewed pages will be located in:

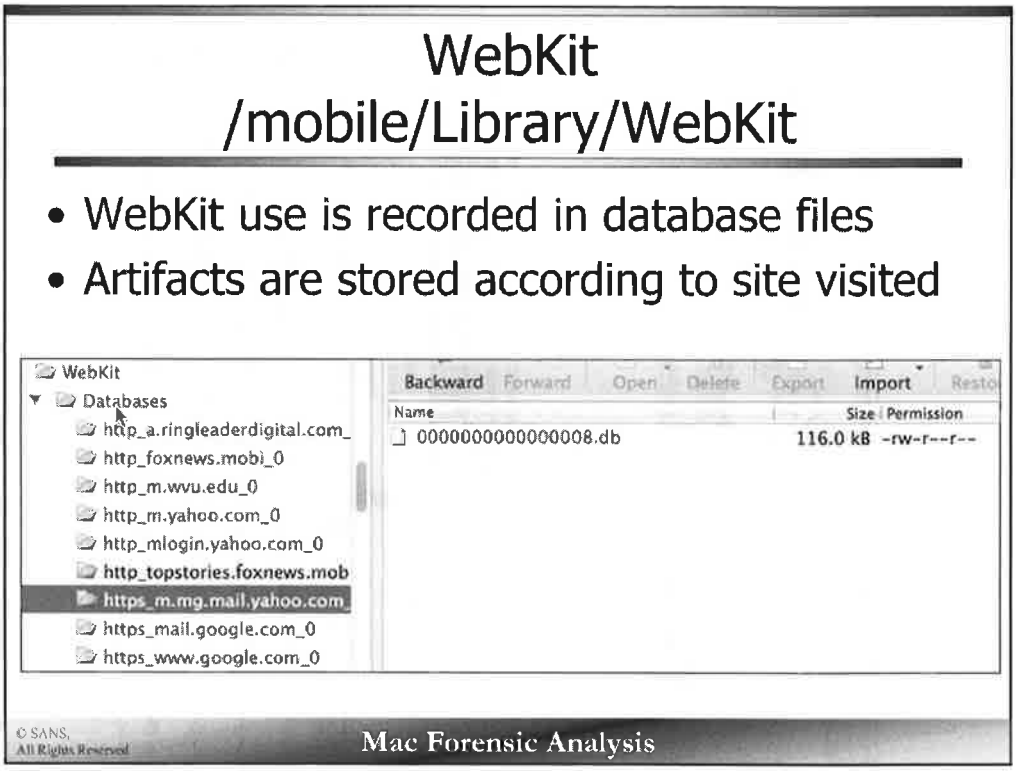
mobile/Library/Caches/Safari/Thumbnail directory in previous iOS versions

In iOS 7.x these same files can be located in:

Mobile/applications/com.apple.mobilesafari/Library/Safari/Thumbnails

Key	Type	Value
Root	Dictionary	(2 items)
▼WebHistoryDates	Array	(4 items)
▼Item 1	Dictionary	(5 items)
	String	http://www.snapchat.com/snakeeyz
▶D	Array	(1 item)
lastVisitedDate	String	387594901.7
title	String	snakeeyz
visitCount	Number	1
▶Item 2	Dictionary	(5 items)

Name	Size	Permission	Date Modified
ED4601A0-0838-4462-9D29-8C659...	56.3 kB	-rw-r--r--	Saturday, April 13, 2013 'PMt' 09:15:03 PM
6A9B9963-55C9-452C-8985-CD7AA...	54.8 kB	-rw-r--r--	Saturday, April 13, 2013 'PMt' 09:14:42 PM
F675D177-1F86-416A-871A-11585E...	55.8 kB	-rw-r--r--	Saturday, April 13, 2013 'PMt' 09:15:16 PM
4CCBCD2B-F30E-4B96-84B4-8AF2F5...	56.5 kB	-rw-r--r--	Saturday, April 13, 2013 'PMt' 09:14:20 PM



WebKits are a cross-platform software, which is responsible for presenting web pages for many widely used web browsers. These Webkits can be found in BlackBerry, Android and iOS devices, and it is currently the underlying technology utilized in Apple's Safari browser.

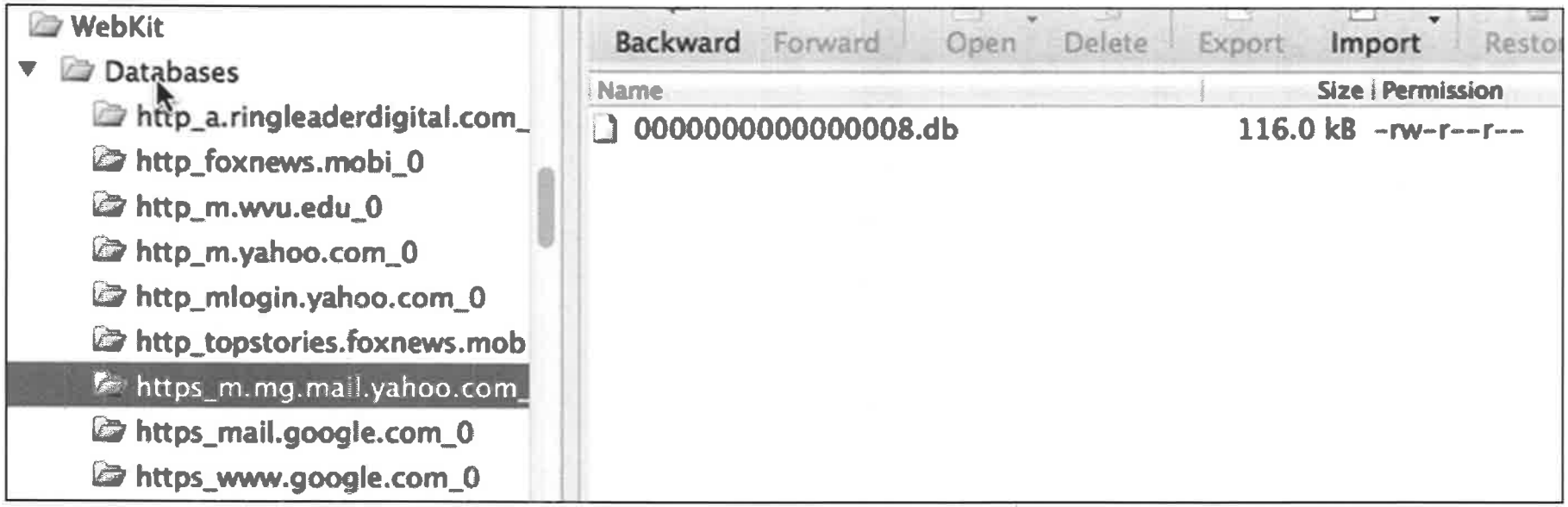
Artifacts in this directory will vary by application, but they are often a result of viewing data from within the web browser. Data is confined to sub-directories representing the site visited, and the bulk of the information is contained in SQLite database files. A common Webkit database file which can offer substantial user data are those associated with web-based e-mail access.

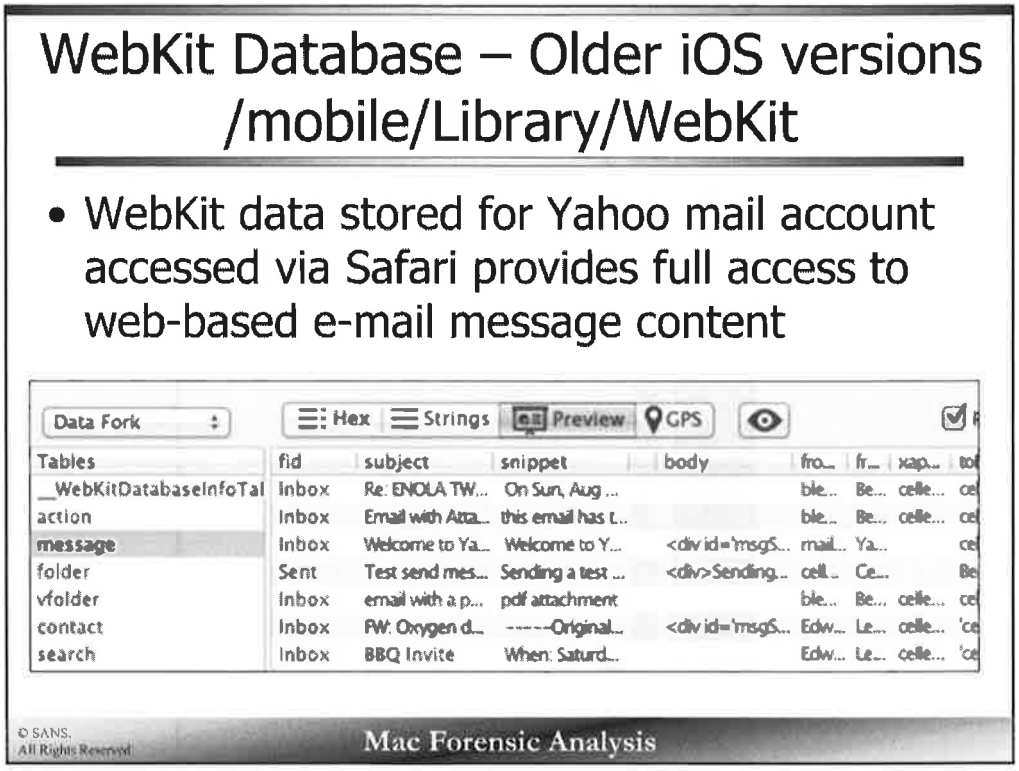
Database files will be located in the path:

mobile/Library/WebKit/Databases/"application" where the application is the name of the site that is visited.

Within this directory, files will use a naming convention like 0000000000000001.db.

Source [1] Zdziarsk, Jonathan. *Hacking and Securing IOS Applications: Stealing Data, Hijacking Software, and How to Prevent it.*

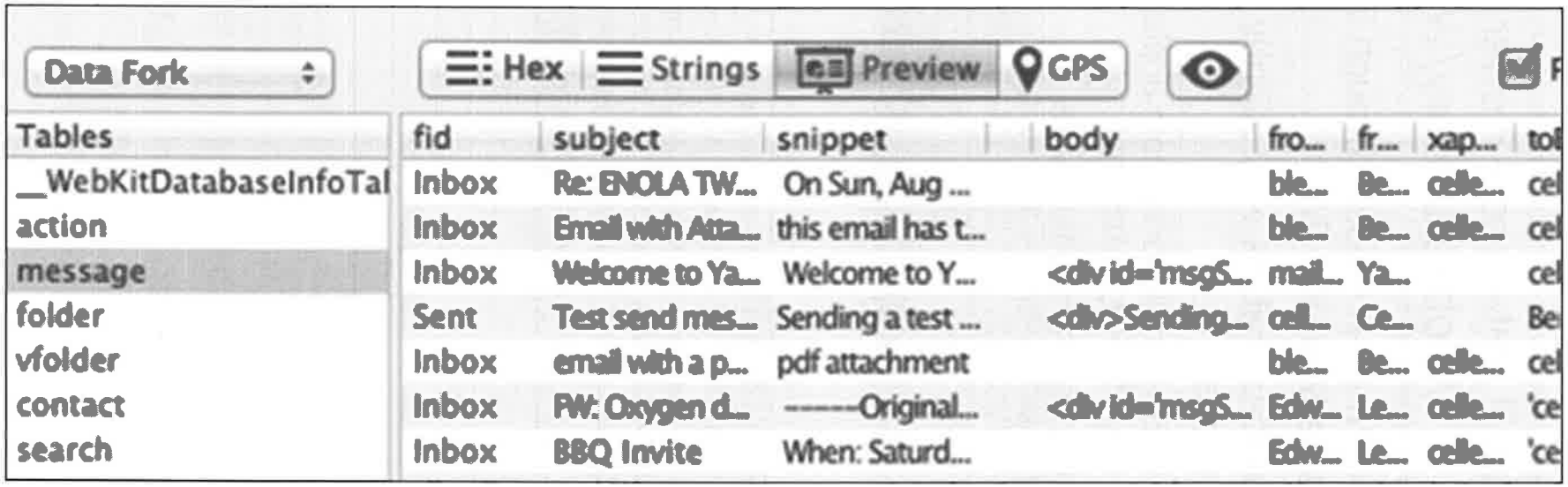

WebKit
Databases
http_a.ringleaderdigital.com_
http_foxnews.mobi_0
http_m.wvu.edu_0
http_m.yahoo.com_0
http_mlogin.yahoo.com_0
http_topstories.foxnews.mob
https_m.mg.mail.yahoo.com_
https_mail.google.com_0
https_www.google.com_0
Backward
Forward
Open
Delete
Export
Import
Name
Size
Permission
0000000000000008.db
116.0 kB
-rw-r--r--



Some tables have been shortened to protect user data but the message table can contain very valuable user data for iOS devices. Look for these important files in mobile/Library/Webkits in older iOS versions.

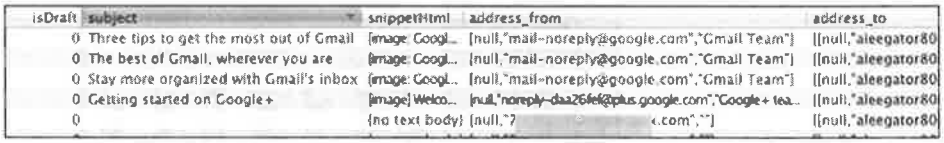
Columns include:

- Message status – is read – 1= read/0 = unread
- Fid – Mailbox location (inbox/sent/draft)
- Subject
- Snippet – partial message content
- Body – full message
- From – e-mail address
- From account
- To address
- CC and BC
- Date received
- Is forwarded - 1 = yes, 0 = No
- Is deleted – 1 = yes, 0 = No
- Has attachment – 1 = yes, 0 = No


Data Fork
Hex
Strings
Preview
GPS
Tables
fid
subject
snippet
body
fro...
fr...
xap...
__WebKitDatabaseInfoTal
action
message
folder
vfolder
contact
search
Inbox
Re: ENOLA TW...
On Sun, Aug ...
Inbox
Email with Atta...
this email has t...
Inbox
Welcome to Ya...
Welcome to Y...
<div id='msgS...
Sent
Test send mes...
Sending a test ...
Sending...
Inbox
email with a p...
pdf attachment
Inbox
FW: Oxygen d...
-----Original...
<div id='msgS...
Inbox
BBQ Invite
When: Saturd...

WebKit Database – Newer iOS versions /mobile/Library/WebKit

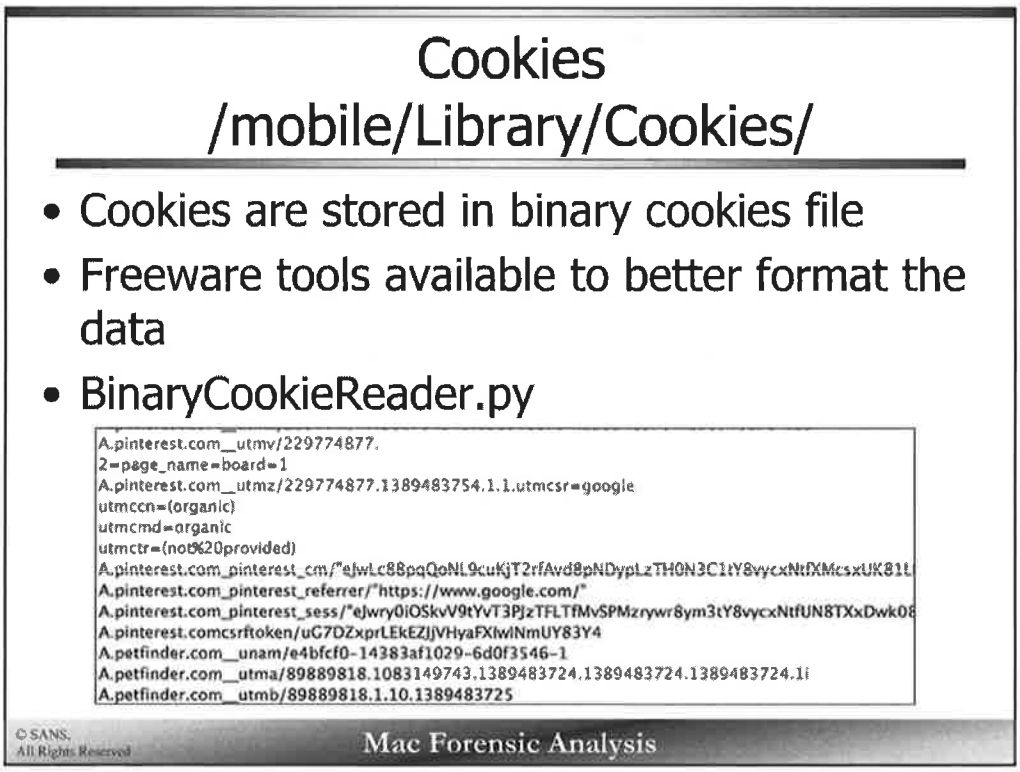
- Web-mail accessed through Safari is available in iOS 7.x
- Database tables vary slightly but the data is similar
  - Mobile/Applications/com.apple.mobilesafari/Library/Webkit/Databases
  - Cached_messages Table



isDraft	subject	snippetHtml	address_from	address_to
0	Three tips to get the most out of Gmail	[image: Googl...	[null,"mail-noreply@google.com","Gmail Team"]	[[null,"aleegator80
0	The best of Gmail, wherever you are	[image: Googl...	[null,"mail-noreply@google.com","Gmail Team"]	[[null,"aleegator80
0	Stay more organized with Gmail's inbox	[image: Googl...	[null,"mail-noreply@google.com","Gmail Team"]	[[null,"aleegator80
0	Getting started on Google+	[image] Welco...	[null,"noreply-daa26fef@plus.google.com","Google+ tea...	[[null,"aleegator80
0		{no text body}	[null,"7 .com",""]	[[null,"aleegator80

© SANS. All Rights Reserved

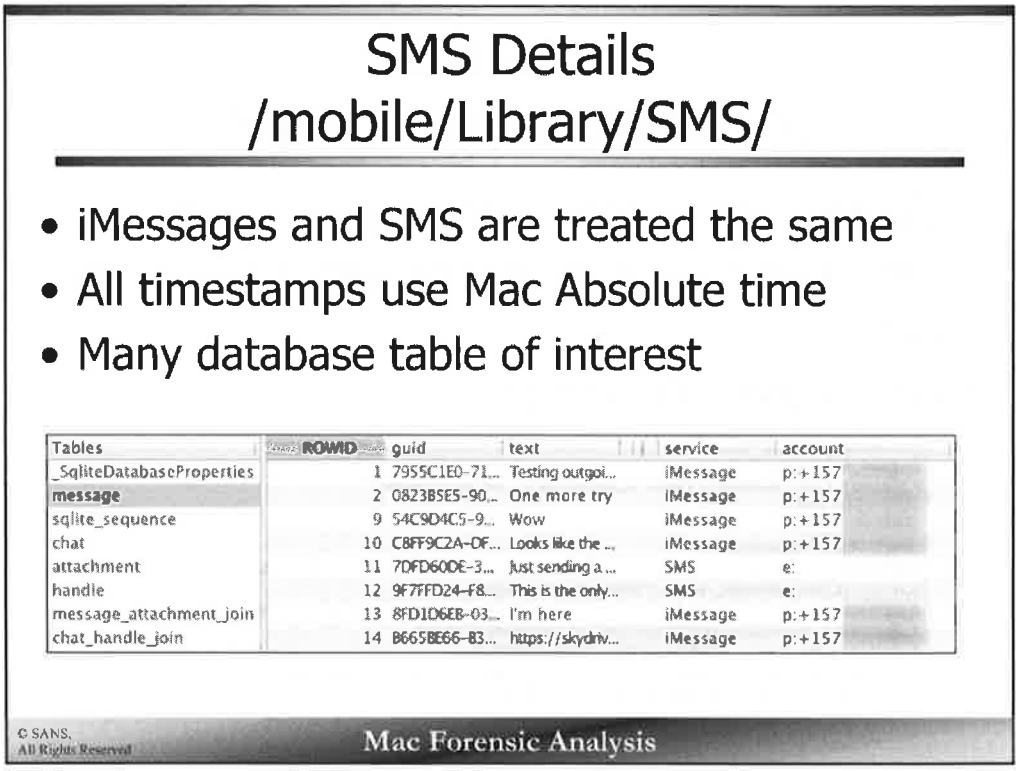
isDraft	subject	snippetHtml	address_from	address_to
0	Three tips to get the most out of Gmail	[image: Googl...	[null,"mail-noreply@google.com","Gmail Team"]	[[null,"aleegator80
0	The best of Gmail, wherever you are	[image: Googl...	[null,"mail-noreply@google.com","Gmail Team"]	[[null,"aleegator80
0	Stay more organized with Gmail's inbox	[image: Googl...	[null,"mail-noreply@google.com","Gmail Team"]	[[null,"aleegator80
0	Getting started on Google+	[image] Welco...	[null,"noreply-daa26fef@plus.google.com","Google+ tea...	[[null,"aleegator80
0		{no text body}	[null,"7 .com",""]	[[null,"aleegator80



Cookies contain pertinent information about a user who is logged into a site at the time that the site was visited. This personal information is saved to make more efficient use of web browsing and often enhances the user experience. Beginning in iOS 5.x, cookie data is now stored in a binary cookies file which varies slightly from the cookies.plist file that was used in previous versions of iOS.

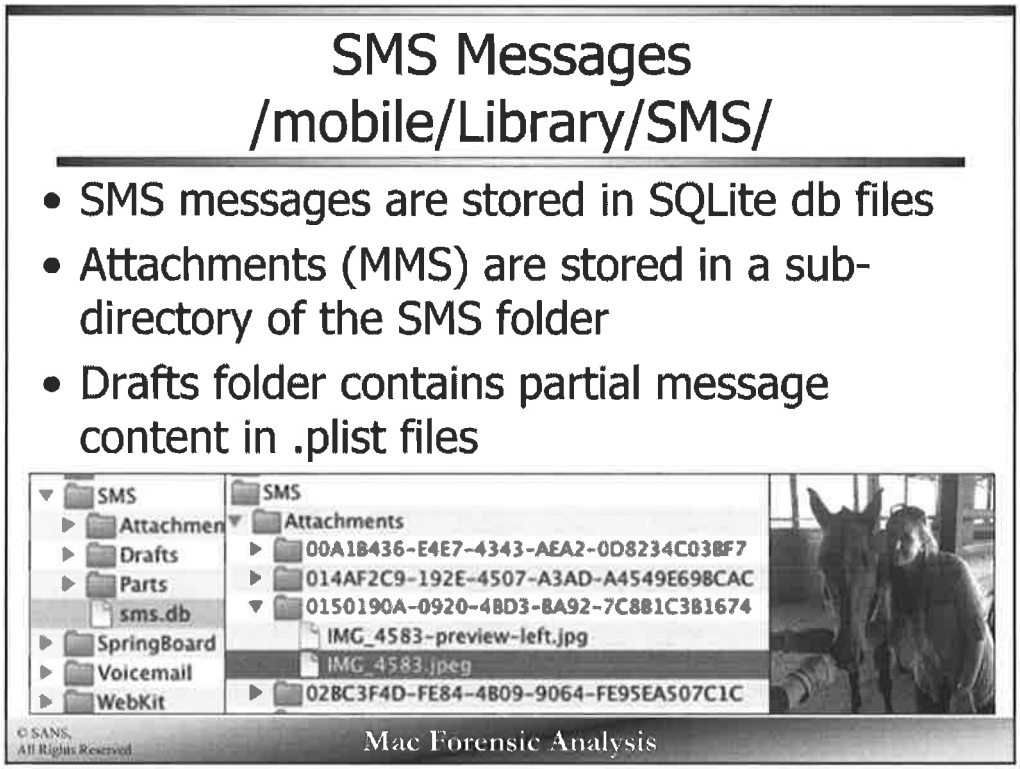
The binarycookiereader.py script is available from http://securitylearn.net/wp-content/uploads/tools/iOS/BinaryCookieReader.py

```
A.pinterest.com__utmv/229774877.
2=page_name=board=1
A.pinterest.com__utmz/229774877.1389483754.1.1.utmcsr=google
utmccn=(organic)
utmcmd=organic
utmctr=(not%20provided)
A.pinterest.com_pinterest_cm/"eJwLc88pqQoNL9cuKjT2rfAvd8pNDypLzTH0N3C1tY8vycxNtfXMcsxUK81L
A.pinterest.com_pinterest_referrer/"https://www.google.com/"
A.pinterest.com_pinterest_sess/"eJwry0iOSkvV9tYvT3PJzTFLTfMvSPMzrywr8ym3tY8vycxNtfUN8TXxDwk0
A.pinterest.comcsrftoken/uG7DZxprLEkEZJjVHyaFXlwlNmUY83Y4
A.petfinder.com__unam/e4bfcf0-14383af1029-6d0f3546-1
A.petfinder.com__utma/89889818.1083149743.1389483724.1389483724.1389483724.1i
A.petfinder.com__utmb/89889818.1.10.1389483725
```



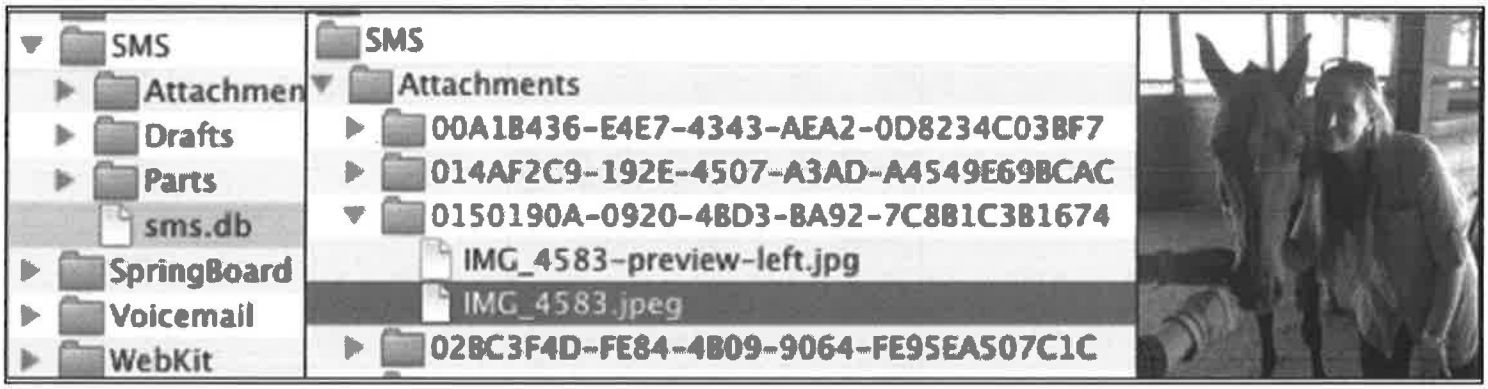
All messages regardless of their type are recorded in Mac Absolute time. iOS 6.0 utilizes the following database tables:

- _SqliteDatabaseProprerties
- message
  - Text – contains the actual message content
  - Service – (SMS versus iMessage)
  - Account – configured account used to send message (iMessages will contain e-mail address or Apple ID)
  - Date – Stored in Mac Absolute time
  - Date Read - Will remain '0' or 'unread' until the message is opened (iMessage)
  - Date Delivered – will only be populated when message is sent as iMessage, will not be populated for SMS
- sqlite_sequence
- chat
  - GUID - contains the recipient/sender information
  - state (incoming, outgoing)
  - Chat identifier – sender/recipient information
  - Service_name - the service used to send the message (SMS versus iMessage)
- attachment
- handle
  - ID - sender/recipient phone number
- message_attachment_join
- chat_handle_join
- chat_message_join



SMS message content is stored in a SQLite database file, sms.db, in the SMS directory. In addition to the sms.db file, this directory may often contain:

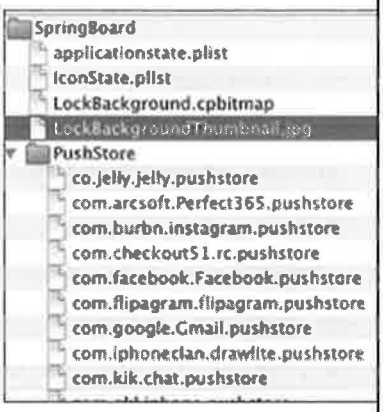
- Attachments sub-directory - contains the image file sent along with the message
- Drafts sub-directory
- Parts sub-directory



Tables	ROWID	guid	text	service	account
_SqliteDatabaseProperties	1	7955C1E0-71...	Testing outgoi...	iMessage	p:+157
message	2	08238SE5-90...	One more try	iMessage	p:+157
sqlite_sequence	9	54C9D4C5-9...	Wow	iMessage	p:+157
chat	10	C8FF9C2A-DF...	Looks like the ...	iMessage	p:+157
attachment	11	7DFD60DE-3...	Just sending a ...	SMS	e:
handle	12	9F7FFD24-F8...	This is the only...	SMS	e:
message_attachment_join	13	8FD1D6EB-03...	I'm here	iMessage	p:+157
chat_handle_join	14	B665BE66-83...	https://skydriv...	iMessage	p:+157

Springboard /mobile/Library/Springboard/

- In iOS 6.x and below the applicationstate.plist was also contained in this directory
- Additional files in the directory
  - IconState.plist
  - LockBackground.cpbitmap
  - LockBackgroundThumbnail.jpg
  - PushStore [Directory]

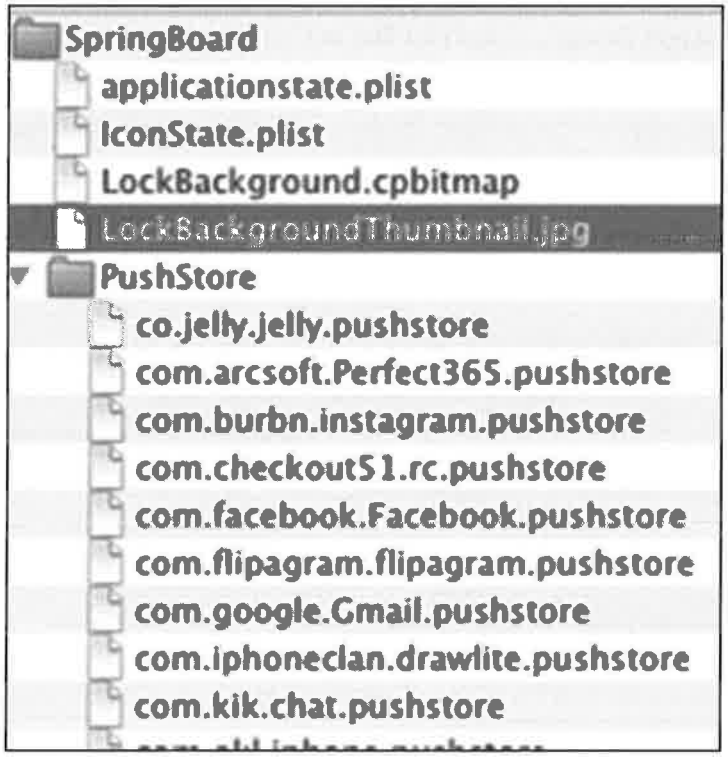


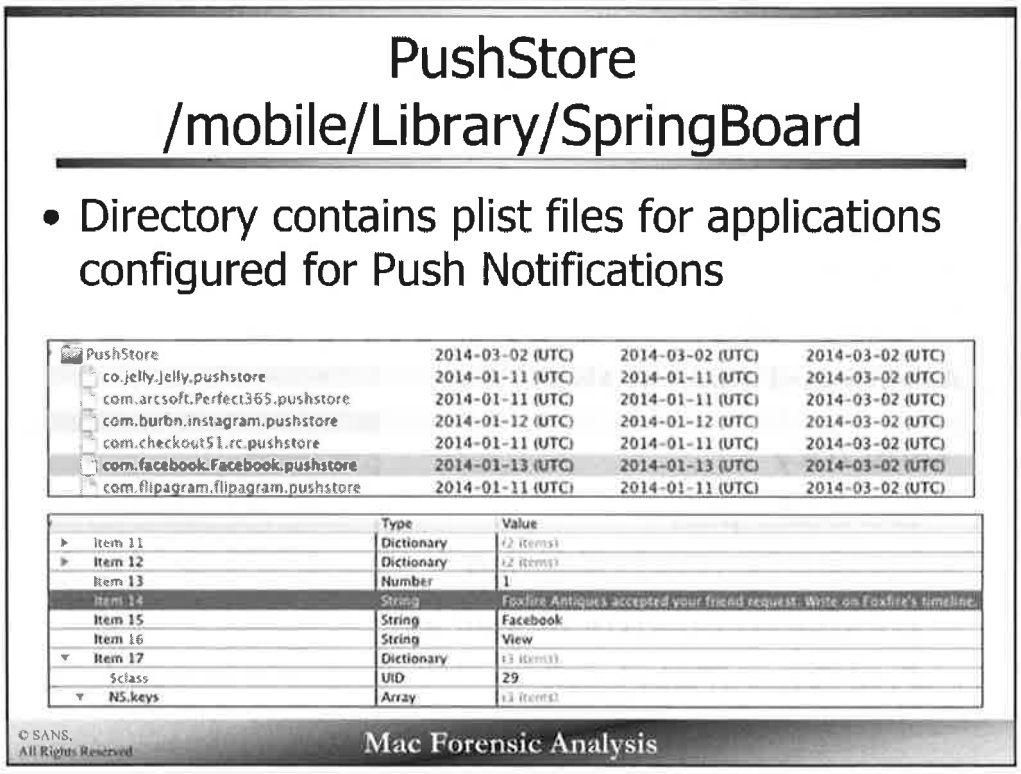
© SANS, All Rights Reserved

IconState.plist is the order/state of the icons at the time of the last reboot. This will tell you how they were arranged.

LockBackground contains an image of the lock screen.

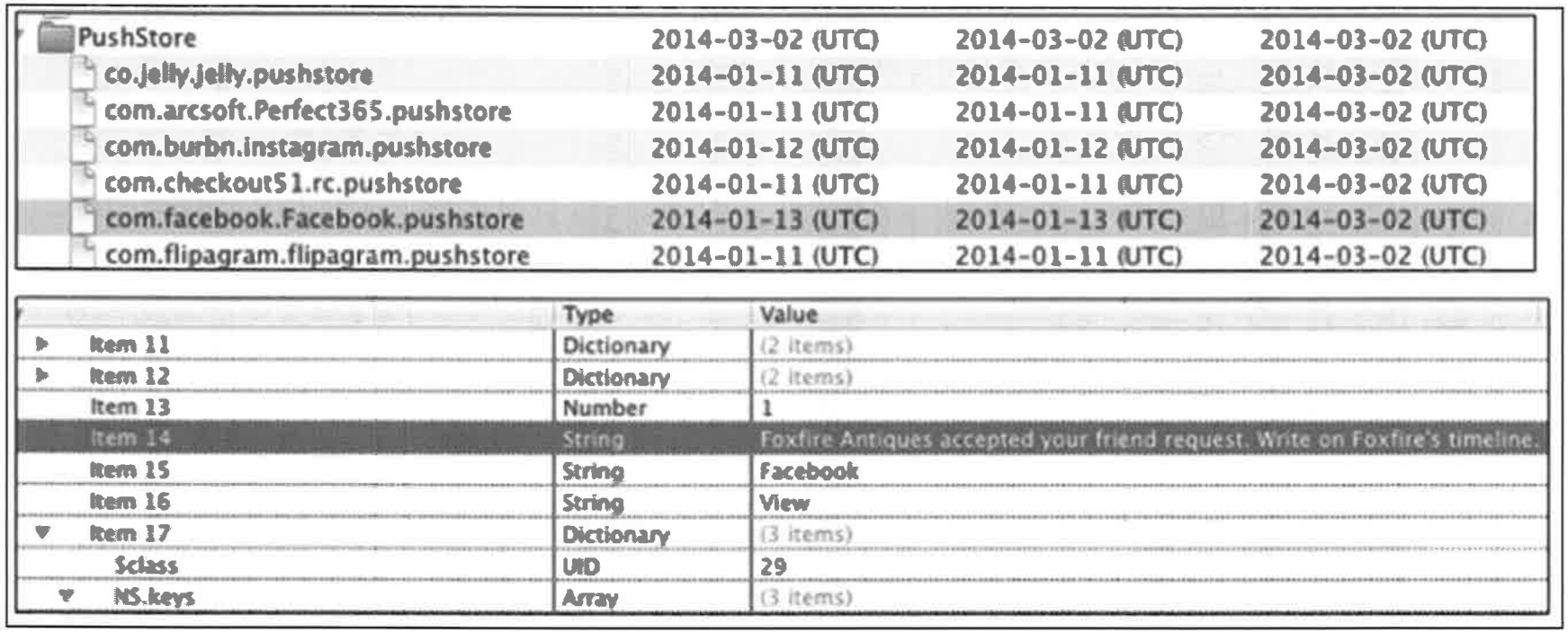
ApplicationState.plist is contained in mobile/Library/BackBoard in iOS 7.x





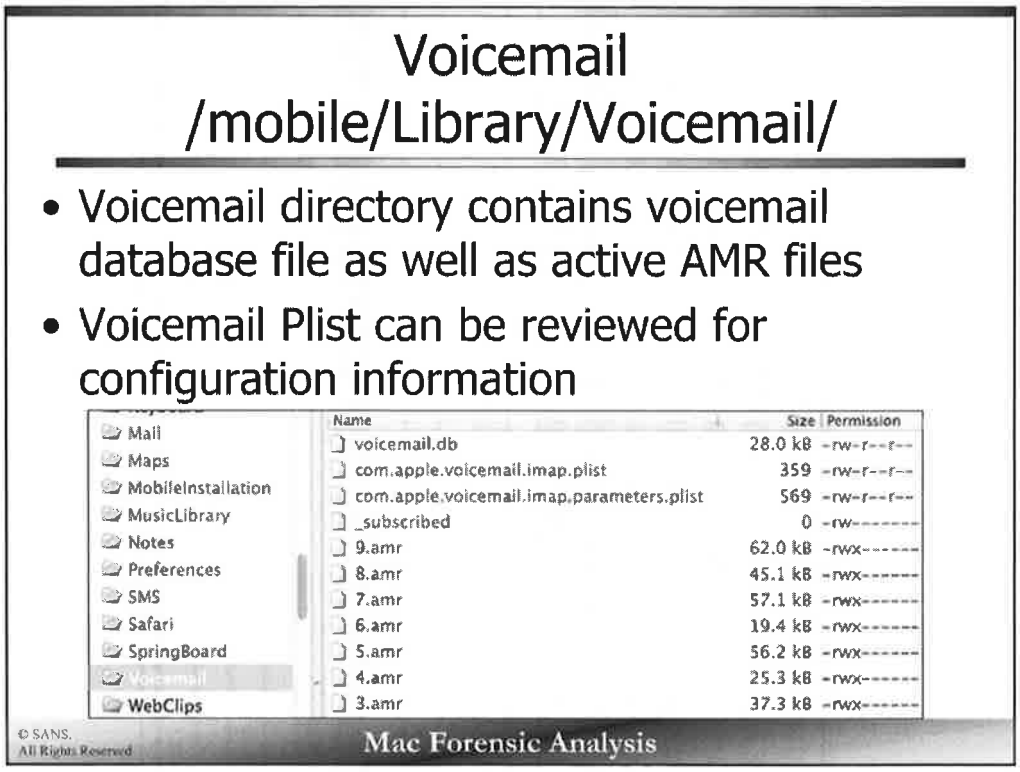
The PushStore Directory is a collection of plist files for each application that was configured to accept Push Notifications. This is an option that is presented to the user upon application installation. Only applications for which the push feature was accepted will appear in this list.

The files in the PushStore Directory MAY contain relevant user data. The example above is an alert that was pushed to the phone via the Facebook application when "Foxfire Antiques" accepted a friend request. Messages sent to the device, will appear throughout these plist files and it may be overlooked by a user that has deleted these items from other locations.



PushStore	2014-03-02 (UTC)	2014-03-02 (UTC)	2014-03-02 (UTC)
co.jelly.jelly.pushstore	2014-01-11 (UTC)	2014-01-11 (UTC)	2014-03-02 (UTC)
com.arcsoft.Perfect365.pushstore	2014-01-11 (UTC)	2014-01-11 (UTC)	2014-03-02 (UTC)
com.burbn.instagram.pushstore	2014-01-12 (UTC)	2014-01-12 (UTC)	2014-03-02 (UTC)
com.checkout51.rc.pushstore	2014-01-11 (UTC)	2014-01-11 (UTC)	2014-03-02 (UTC)
com.facebook.Facebook.pushstore	2014-01-13 (UTC)	2014-01-13 (UTC)	2014-03-02 (UTC)
com.flipagram.flipagram.pushstore	2014-01-11 (UTC)	2014-01-11 (UTC)	2014-03-02 (UTC)

	Type	Value
▶ Item 11	Dictionary	(2 items)
▶ Item 12	Dictionary	(2 items)
Item 13	Number	1
Item 14	String	Foxfire Antiques accepted your friend request. Write on Foxfire's timeline.
Item 15	String	Facebook
Item 16	String	View
▼ Item 17	Dictionary	(3 items)
$class	UID	29
▼ NS.keys	Array	(3 items)



The Voicemail directory located in "mobile/Library/Voicemail/ typically contains a voicemail plist file, "com.apple.voicemail.imap.parameters.plist", a "voicemail.db" file, and multiple AMR files containing the audio recording of messages retained on the device.

com.apple.voicemail.imap.parameters.plist will contain voicemail configuration settings to include:

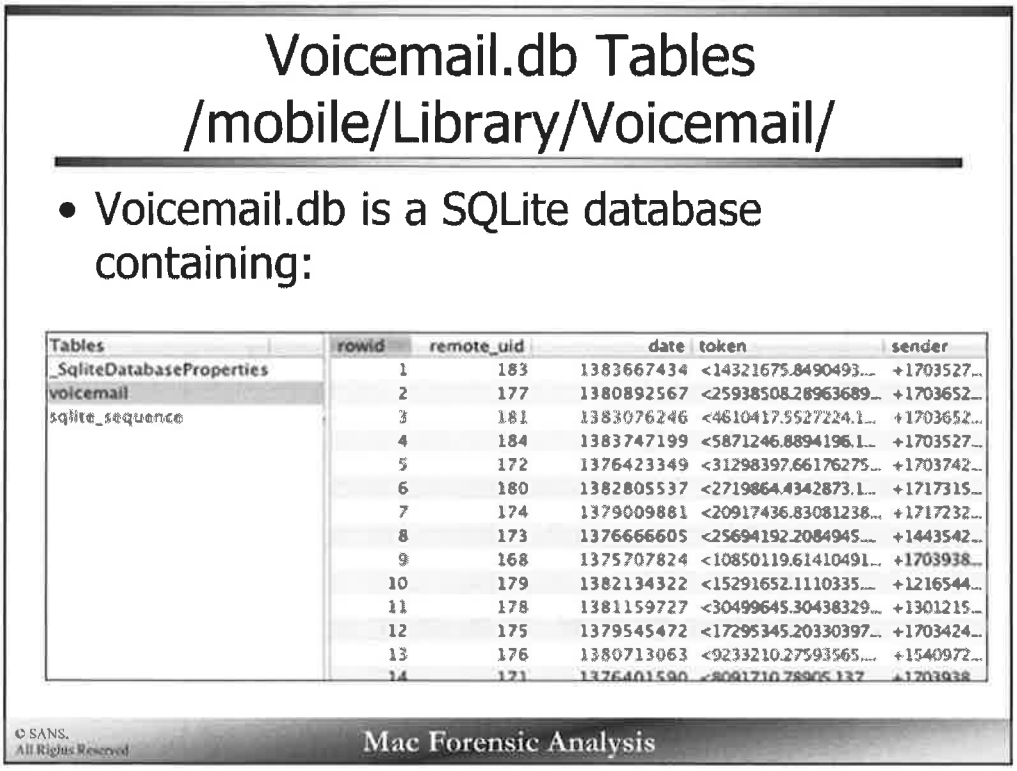
- the number set up to receive voicemail messages
- the port through which messages are received
- the networked server
- whether or not the account is syncing

The actual audio files themselves are saved in the Adaptive Multi Rate, AMR, file format. AMR is the standard for many mobile devices and VoIP technologies. AMR audio files are relatively small and of good quality. They can be played using the QuickTime Player, which comes installed by default with Mac OS, but they can also be converted to other audio formats to include: WAV, MPA, and MP3.

*Cellular providers who require users to log into their system to access voicemail may not have access to cached AMR files stored on the device.

For more information on the types of audio and video files supported by Mac OS, consult the following link:

Source: [1] http://support.apple.com/kb/HT3775?viewlocale=en_US&locale=en_US



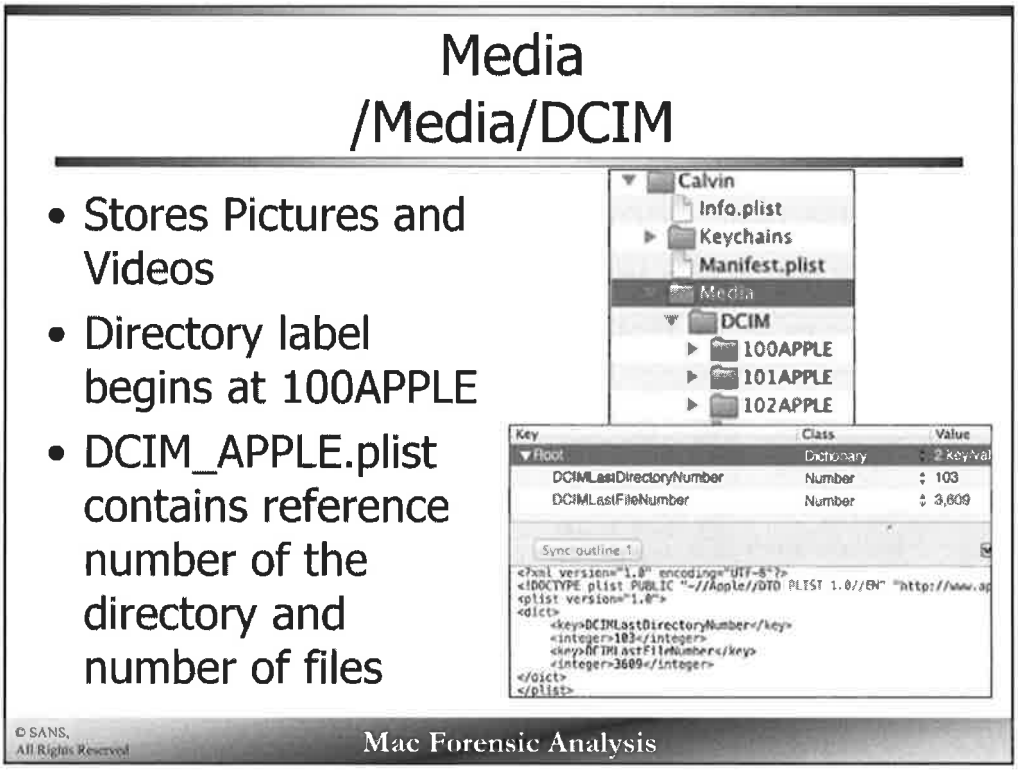
The voicemail.db file will contain three tables:

- _SqliteDatabaseProperties
- sqlite_sequence
- voicemail

The voicemail table contains the most data of interest. The table contains multiple columns for storing important items like:

- Date
- Sender

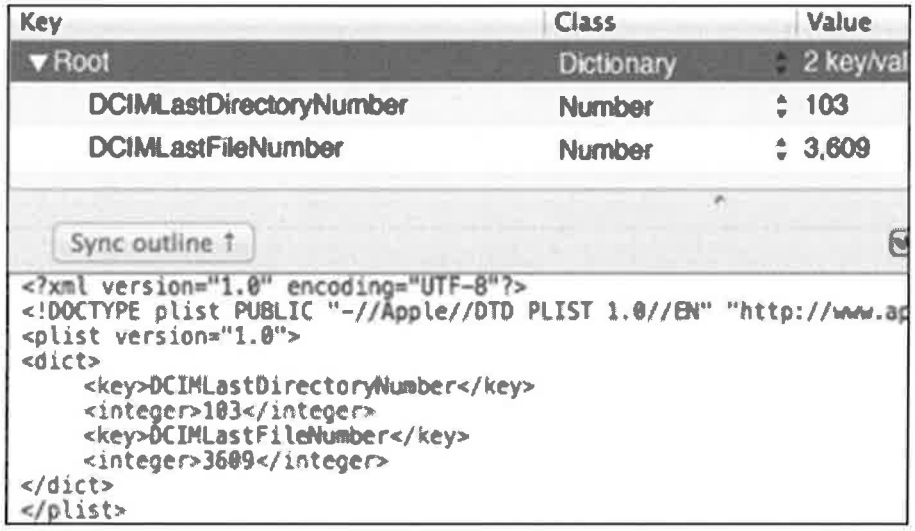
Tables	rowid	remote_uid	date	token	sender
_SqliteDatabaseProperties	1	183	1383667434	<14321675.8490493....	+1703527...
voicemail	2	177	1380892567	<25938508.28963689...	+1703652...
sqlite_sequence	3	181	1383076246	<4610417.5527224.1...	+1703652...
	4	184	1383747199	<5871246.8894196.1...	+1703527...
	5	172	1376423349	<31298397.66176275...	+1703742...
	6	180	1382805537	<2719864.4342873.1...	+1717315...
	7	174	1379009881	<20917436.83081238...	+1717232...
	8	173	1376666605	<25694192.2084945....	+1443542...
	9	168	1375707824	<10850119.61410491...	+1703938...
	10	179	1382134322	<15291652.1110335....	+1216544...
	11	178	1381159727	<30499645.30438329...	+1301215...
	12	175	1379545472	<17295345.20330397...	+1703424...
	13	176	1380713063	<9233210.27593565....	+1540972...
	14	171	1376401590	<8091710.78905.137	+1703938

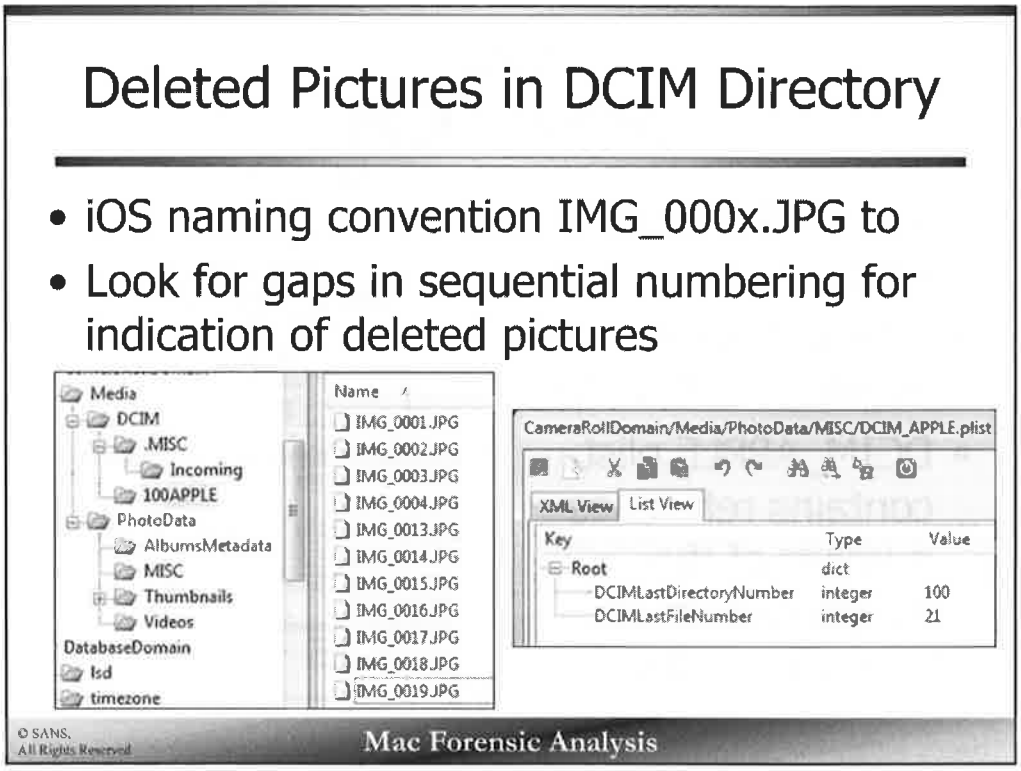


The Media/DCIM folder is the directory used to store images and video files captured with the iDevice's built-in camera. Below the main DCIM directory you will find sub-directories starting with the label, "100APPLE". Each APPLE sub-directory can hold a maximum of 999 files. When the directory reaches 1000, a new sub-directory is created using an incremented number. When image 1001 is taken on the iOS device, a new directory, 101APPLE is created, and an entry is made in the file, "DCIM_APPLE.plist notating the addition. This process is recreated each time the directory reaches the maximum of 1000 files.

The data that is tracked by DCIM_APPLE.plist includes:

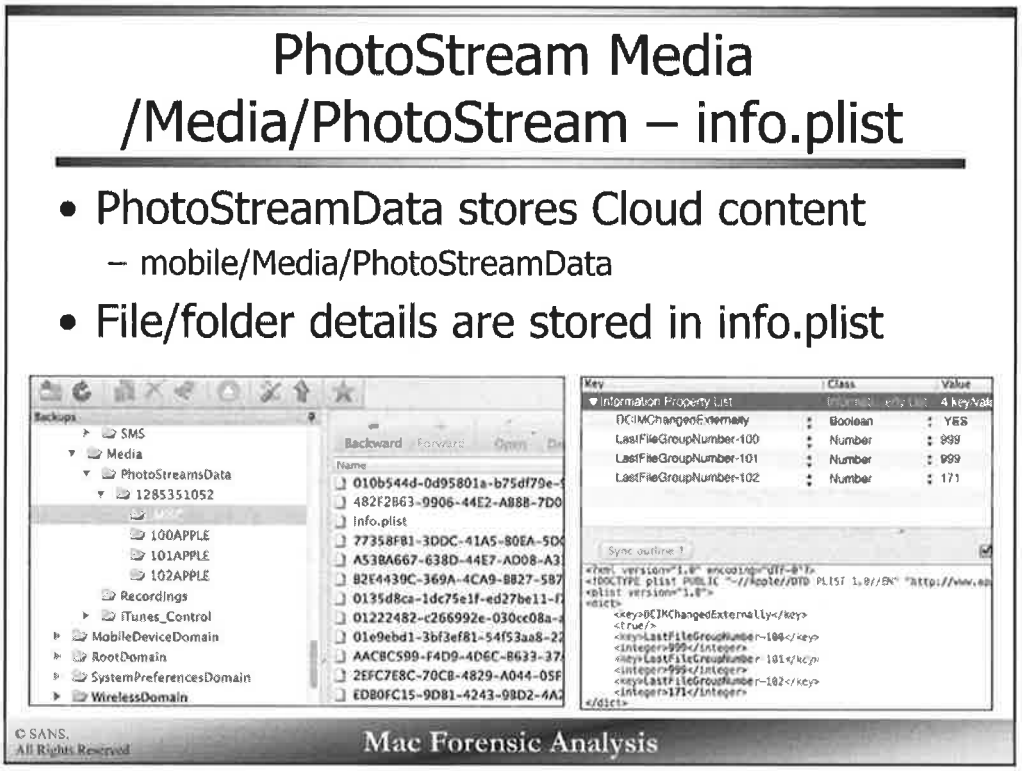
DCIMLastDirectoryNumber</key>
<integer>103</integer>
<key>DCIMLastFileNumber</key>
<integer>3609</integer>





The iOS standard for naming image files captured using the device camera is IMG_000x.JPG where the numbering begins at 1 and increases each time a new image is captured. Look for gaps in the numbering sequence as evidence of deleted images that could be retained elsewhere. A quick review of the contents of the DCIM directory above shows that IMG_0005.JPG through IMG_0012.JPG have been deleted.

Although it may not be apparent at first, IMG_0019.JPG and IMG_0020.JPG have also been deleted, which can be confirmed by the DCIMLastFileNumber of 21 which is retained in the DCIM_APPLE.plist file.



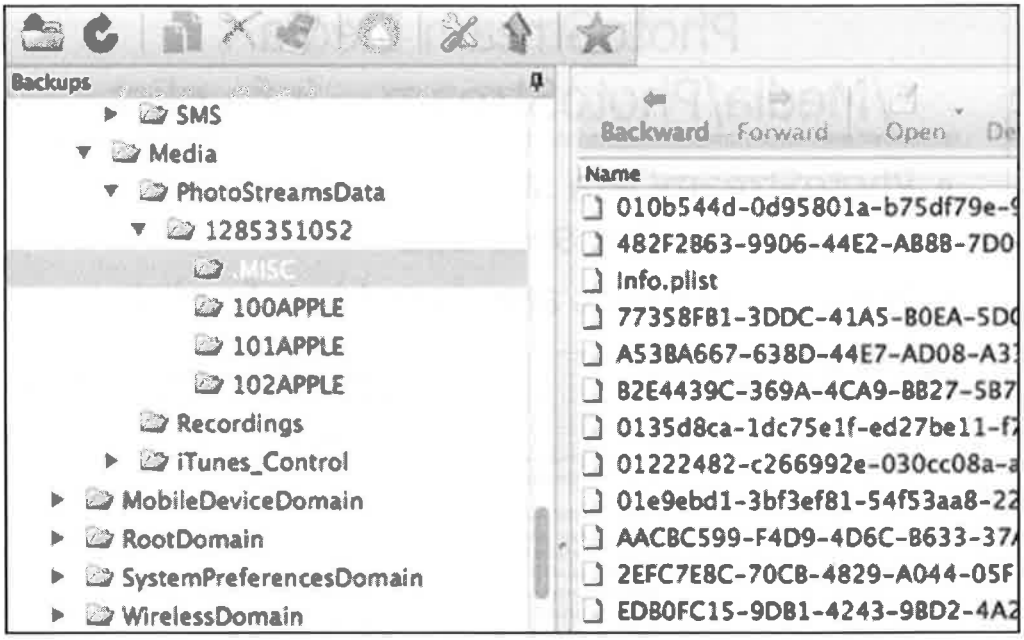
File Types

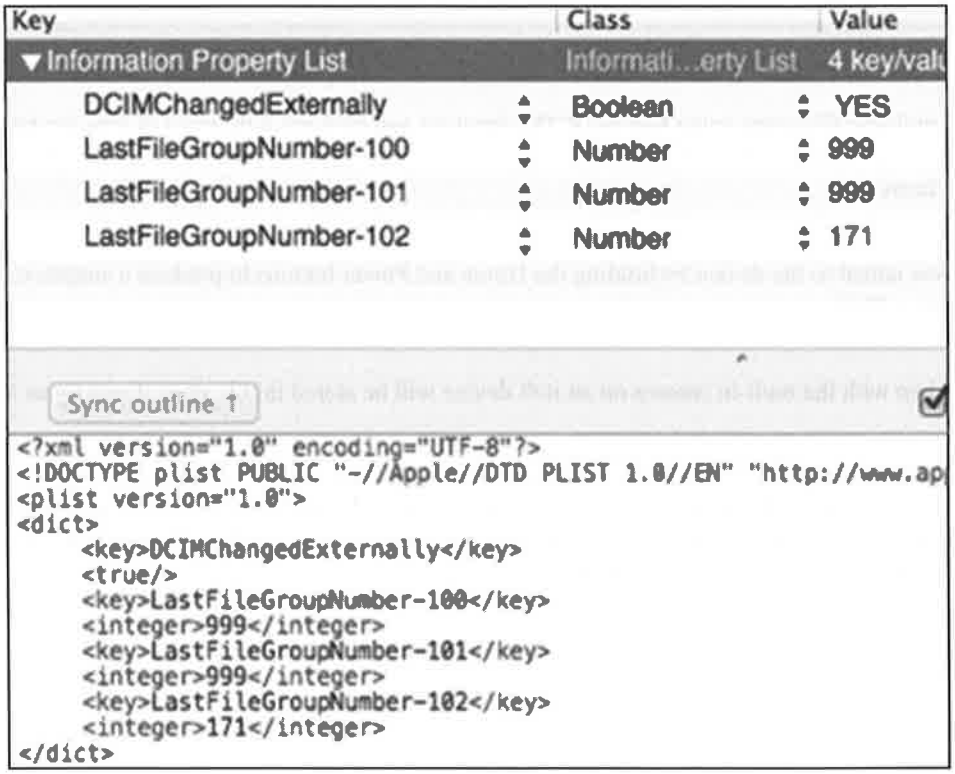
You will find multiple file types in the Media/DCIM directory and each are a function of how the image was captured/saved on the device. Images captured using the built-in camera application on an iOS device will be stored in JPG format.

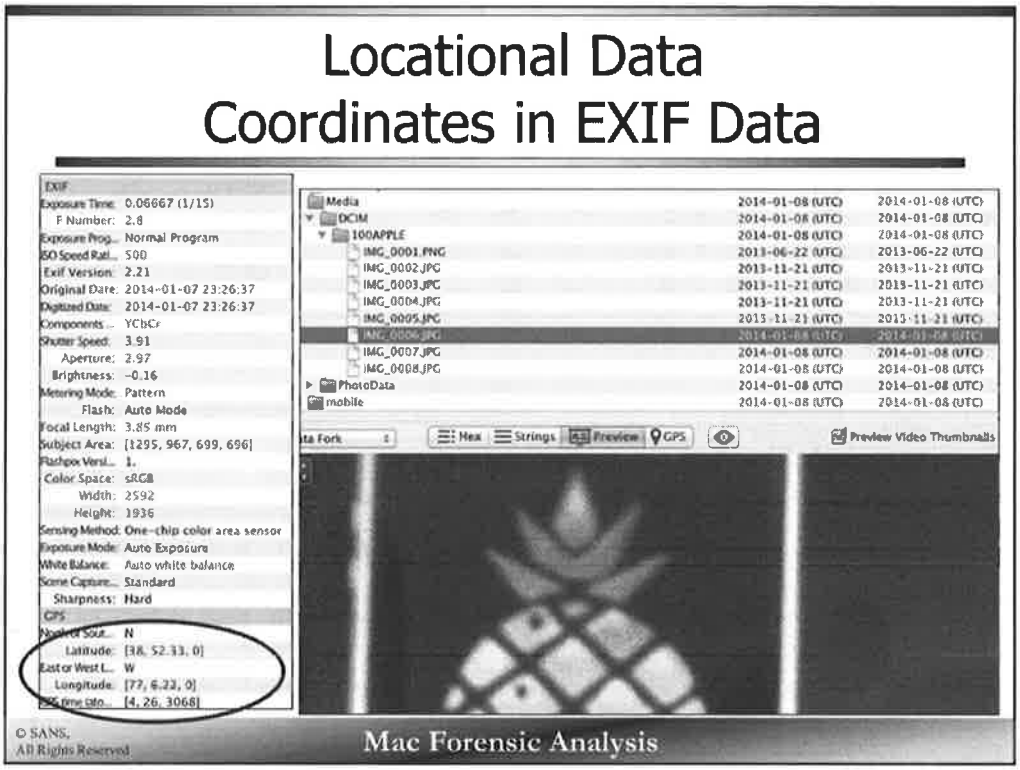
If the image was added to the device by holding the Home and Power buttons to produce a snapshot, this image will be stored as a PNG.

Movie clips taken with the built-in camera on an iOS device will be stored in this same directory as MOV files.

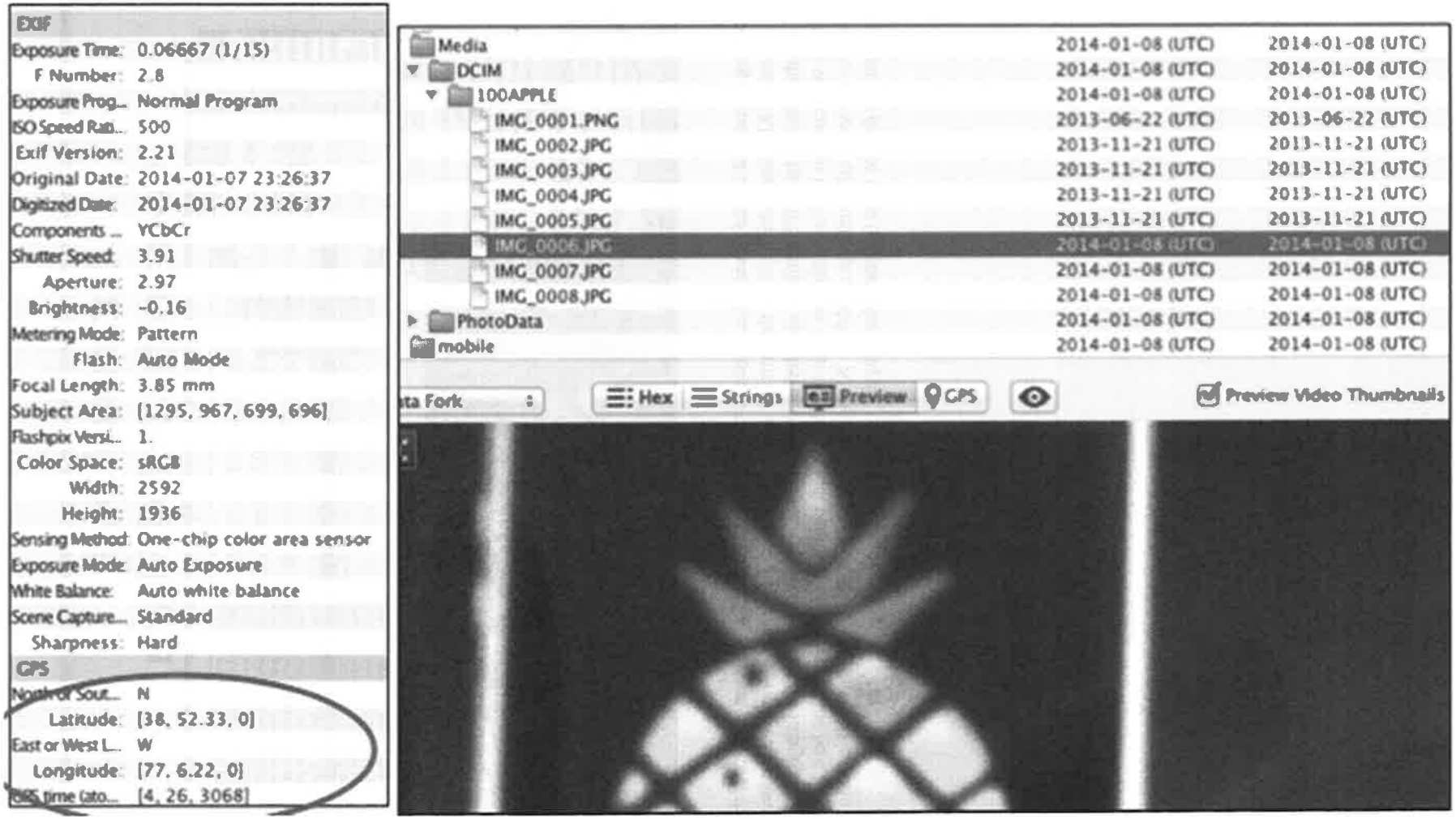
The same file storage formats are applied to the data stored in the Photo Stream directory (mobile/Media/PhotoStreamData). These images are synced when the device is connected and logged into iTunes. The configuration file containing the information about the files in the PhotoStreamData folder is stored in an info.plist file in the ".MISC" sub-directory.

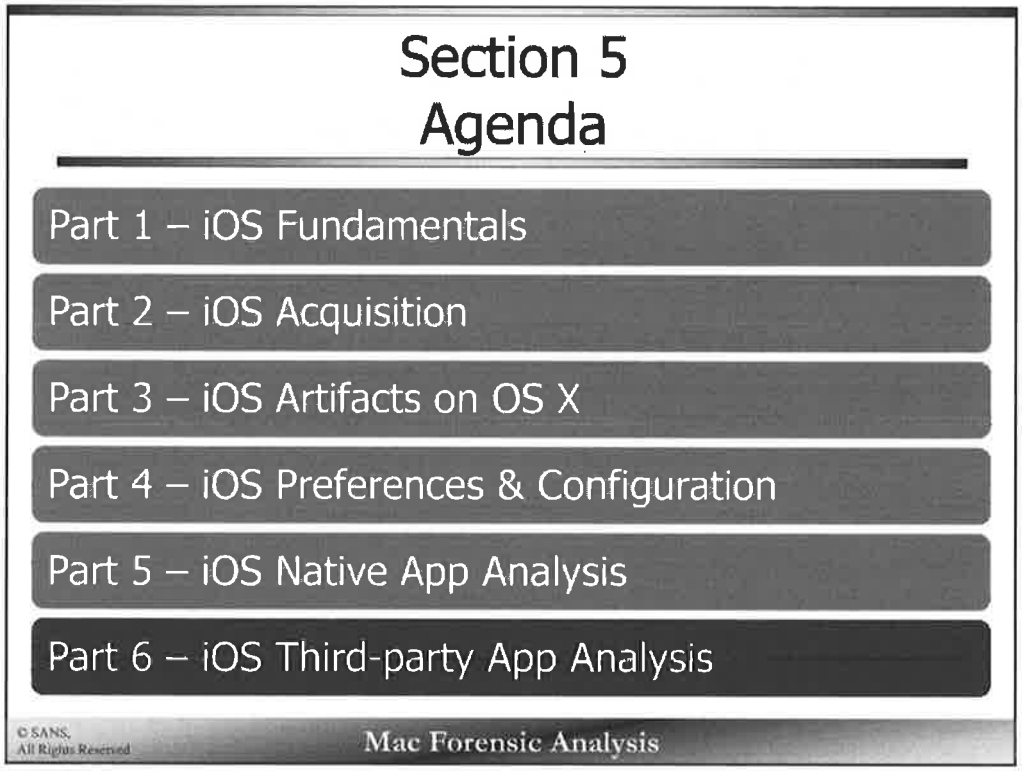

Backups
SMS
Media
PhotoStreamsData
1285351052
.MISC
100APPLE
101APPLE
102APPLE
Recordings
iTunes_Control
MobileDeviceDomain
RootDomain
SystemPreferencesDomain
WirelessDomain
Backward
Forward
Open
Name
010b544d-0d95801a-b75df79e-
482F2863-9906-44E2-AB8B-7D0
Info.plist
77358FB1-3DDC-41A5-B0EA-5D
A53BA667-638D-44E7-AD08-A3
B2E4439C-369A-4CA9-BB27-5B7
0135d8ca-1dc75e1f-ed27be11-f
01222482-c266992e-030cc08a-
01e9ebd1-3bf3ef81-54f53aa8-22
AACBC599-F4D9-4D6C-B633-37
2EFC7E8C-70CB-4829-A044-05F
EDB0FC15-9DB1-4243-98D2-4A


Key
Class
Value
Information Property List
Informati...erty List
4 key/val
DCIMChangedExternally
Boolean
YES
LastFileGroupNumber-100
Number
999
LastFileGroupNumber-101
Number
999
LastFileGroupNumber-102
Number
171
Sync outline 1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.ap
<plist version="1.0">
<dict>
<key>DCIMChangedExternally</key>
<true/>
<key>LastFileGroupNumber-100</key>
<integer>999</integer>
<key>LastFileGroupNumber-101</key>
<integer>999</integer>
<key>LastFileGroupNumber-102</key>
<integer>171</integer>
</dict>

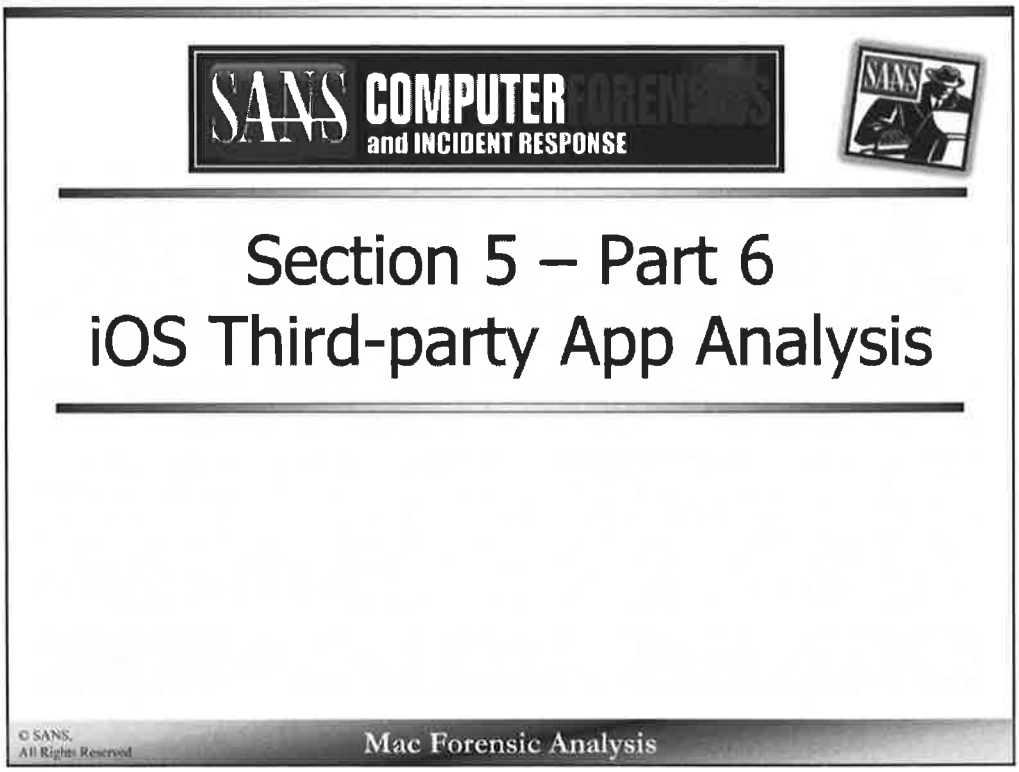


If the GPS setting is enabled on the device (i.e. Settings>>Privacy>>Location Services set to "On"), these images will contain EXIF data, which stores the latitude and longitude of the device at the time the image was captured. If another device had GPS settings enabled and was used to capture and send the image, then the incoming image will contain GPS coordinates from the device that sent the image even if the host that received the image had the device's GPS settings disabled. Along with coordinates, the EXIF data will contain information about the device used to create the image.


EXIF
Exposure Time: 0.06667 (1/15)
F Number: 2.8
Exposure Prog... Normal Program
ISO Speed Rati... 500
Exif Version: 2.21
Original Date: 2014-01-07 23:26:37
Digitized Date: 2014-01-07 23:26:37
Components ... YCbCr
Shutter Speed: 3.91
Aperture: 2.97
Brightness: -0.16
Metering Mode: Pattern
Flash: Auto Mode
Focal Length: 3.85 mm
Subject Area: [1295, 967, 699, 696]
Flashpix Versi... 1.
Color Space: sRGB
Width: 2592
Height: 1936
Sensing Method: One-chip color area sensor
Exposure Mode: Auto Exposure
White Balance: Auto white balance
Scene Capture... Standard
Sharpness: Hard
GPS
North or Sout... N
Latitude: [38, 52.33, 0]
East or West L... W
Longitude: [77, 6.22, 0]
GPS time (ato... [4, 26, 3068]
Media 2014-01-08 (UTC) 2014-01-08 (UTC)
DCIM 2014-01-08 (UTC) 2014-01-08 (UTC)
100APPLE 2014-01-08 (UTC) 2014-01-08 (UTC)
IMG_0001.PNG 2013-06-22 (UTC) 2013-06-22 (UTC)
IMG_0002.JPG 2013-11-21 (UTC) 2013-11-21 (UTC)
IMG_0003.JPG 2013-11-21 (UTC) 2013-11-21 (UTC)
IMG_0004.JPG 2013-11-21 (UTC) 2013-11-21 (UTC)
IMG_0005.JPG 2013-11-21 (UTC) 2013-11-21 (UTC)
IMG_0006.JPG 2014-01-08 (UTC) 2014-01-08 (UTC)
IMG_0007.JPG 2014-01-08 (UTC) 2014-01-08 (UTC)
IMG_0008.JPG 2014-01-08 (UTC) 2014-01-08 (UTC)
PhotoData 2014-01-08 (UTC) 2014-01-08 (UTC)
mobile 2014-01-08 (UTC) 2014-01-08 (UTC)
ta Fork
Hex
Strings
Preview
GPS
Preview Video Thumbnails



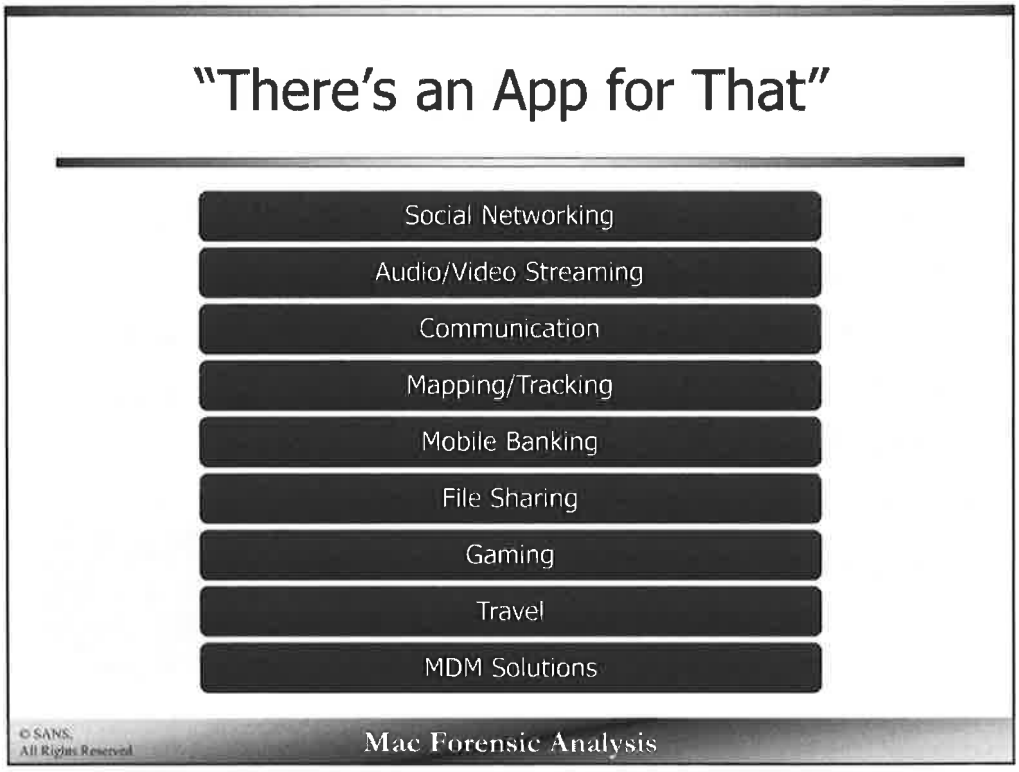
This page intentionally left blank.



This page intentionally left blank.

Third-party applications are those that are not native to the Apple operating system but are engineered to run on this and other platforms. Third-party applications can be downloaded from Apple's iTunes store. With over 500 million users subscribing to iTunes, it makes sense that the iOS platform would offer the most downloadable applications for its consumers. Numbers exceed one million available applications and that number is growing every day.

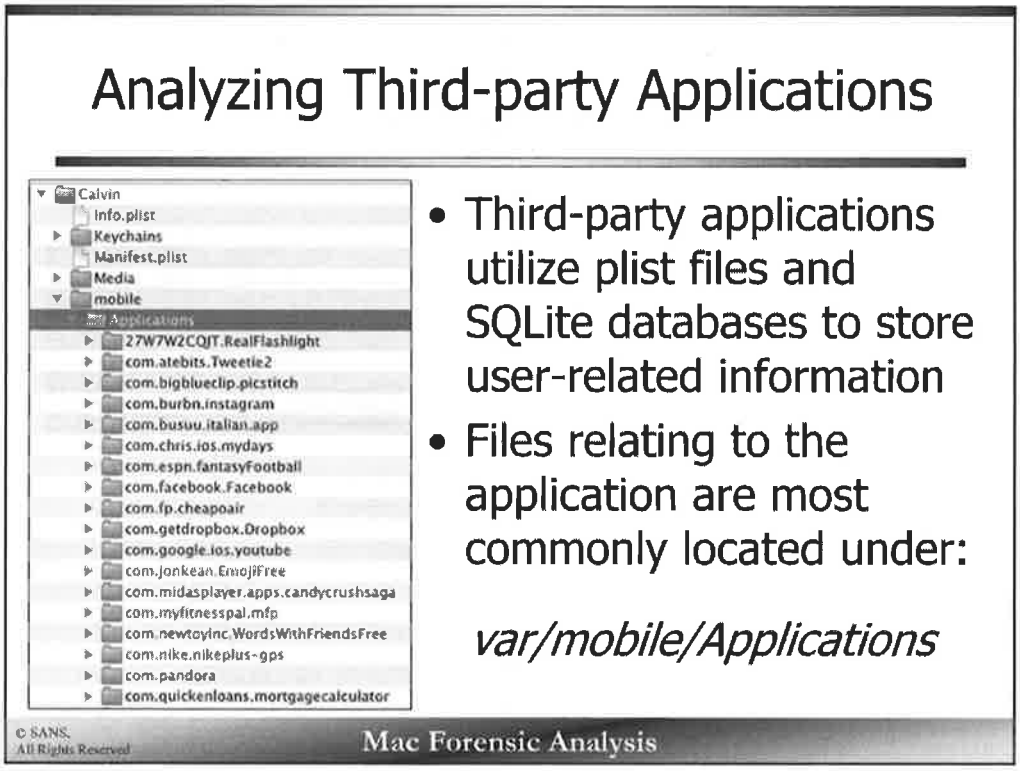
The majority of commercial mobile forensic tools and freeware alike will display third-party applications in a directory that is separate from the native iOS applications. Look for third-party applications in the **Mobile/Applications** directory in BlackLight and under **User App Files** in iBackupBot.



Nearly any task imaginable has an associated application to make the process more efficient on an iOS device. Apple has recently categorized their applications into the following subsets:

- Books
- Business
- Catalogs
- Education
- Entertainment
- Finance
- Food/Drink
- Games
- Health & Fitness
- Kids
- Lifestyle
- Medical
- Music
- Navigation
- News
- Newsstand
- Photo/Video
- Productivity
- Reference
- Social Networking
- Sports
- Travel
- Utilities
- Weather

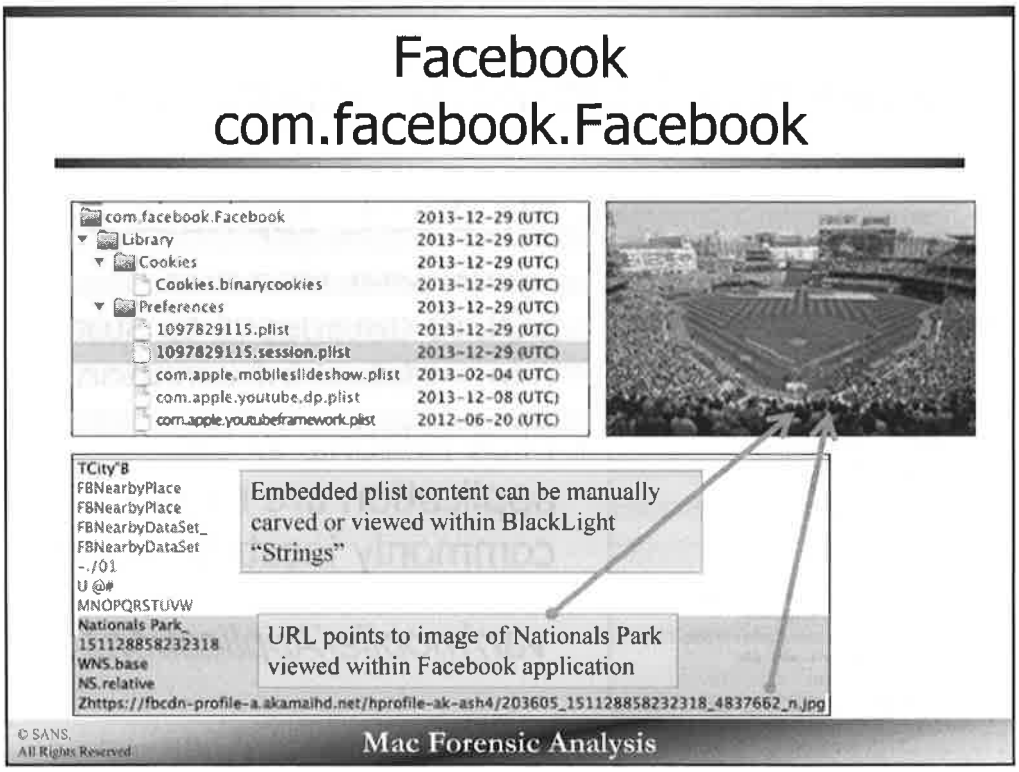
Third-party applications are usually grouped by their functionality; however, most applications perform a variety of functions—some of which can be completely overlooked by the user.



Application data will be stored in its respective folder under the var/mobile/Applications.

It is important to note that when examining these files natively on a Mac, no special tools are needed. BlackLight was utilized to examine the files acquired during an iTunes backup; however, these files can be viewed by exporting the contents and viewing the files with a Property List (plist) editor, a free Hex editor and/or a SQLite browsing utility.

The typical structure for this application data is a folder containing the application and subfolders for "Documents" and "Library" files. Nested in these folders are "Preferences", "Caches" and any number of log files, plists and database files containing application information.

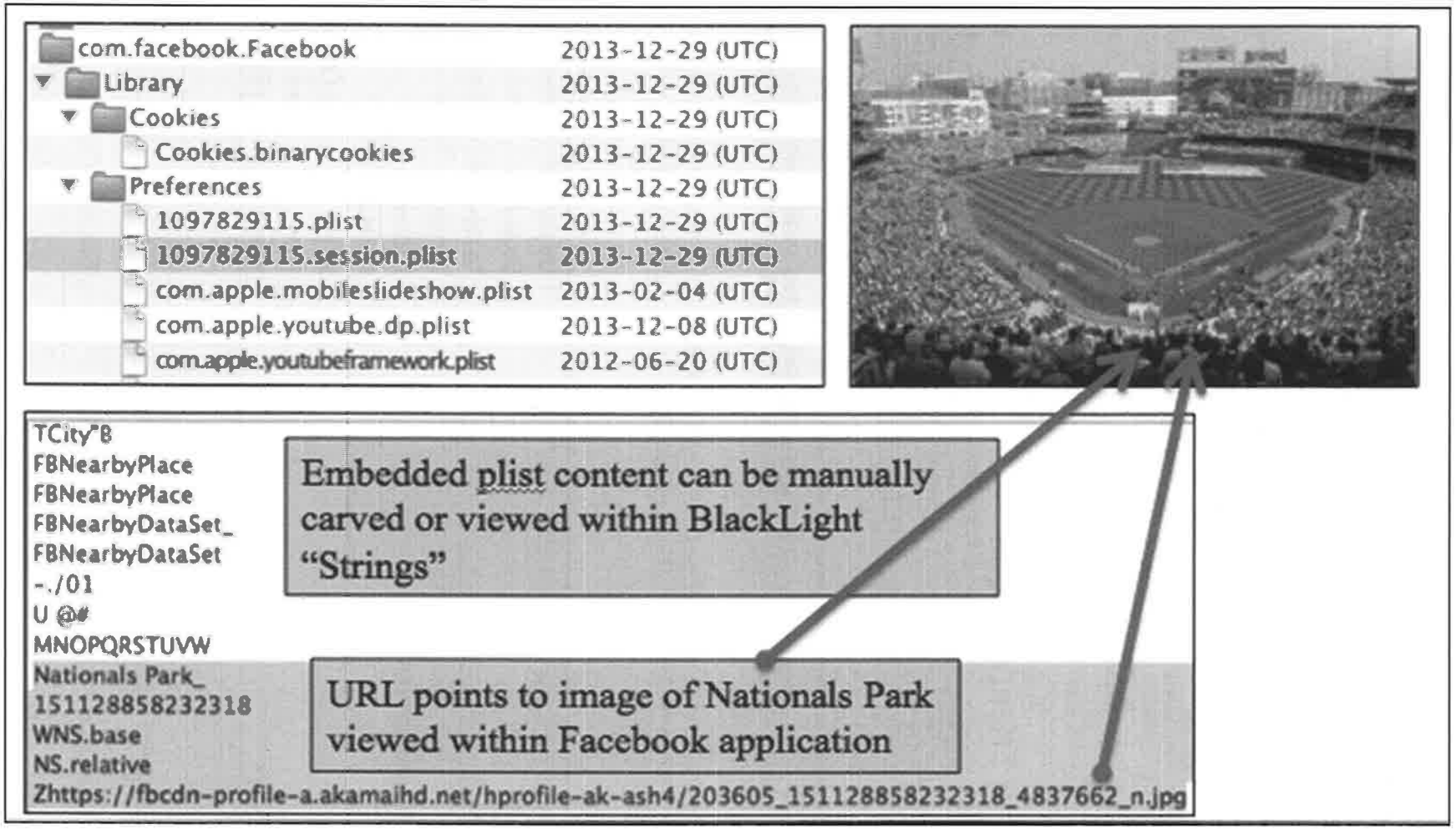


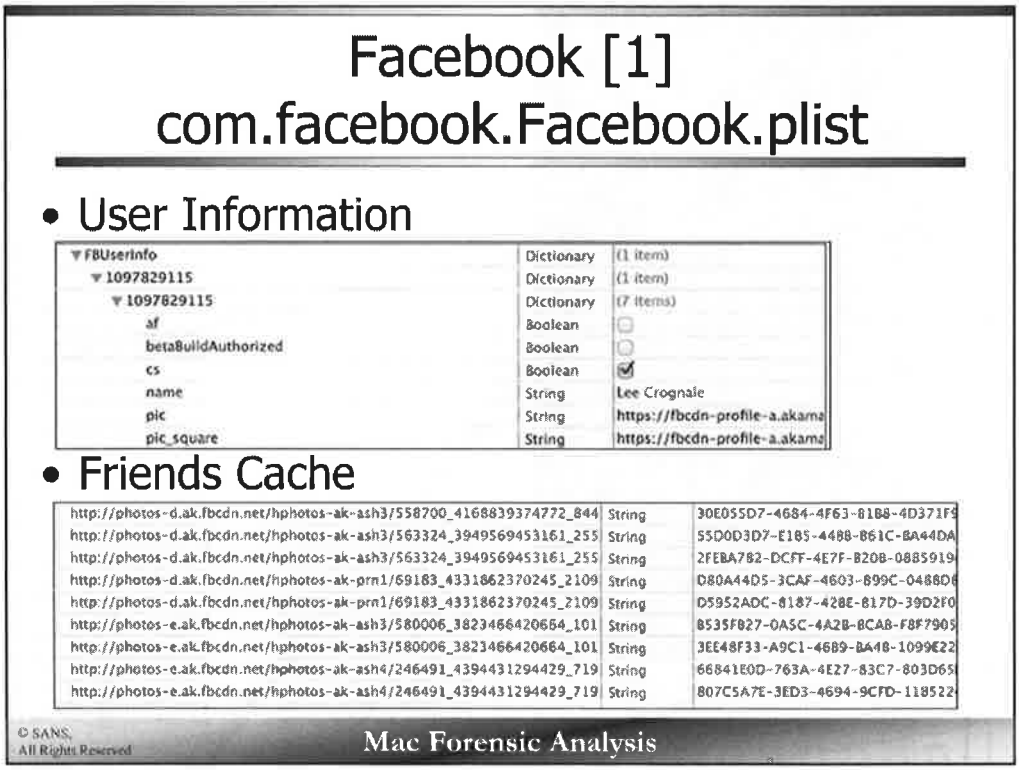
Facebook is a very common third-party application.

Under the application folder, com.facebook.Facebook, there are many user related files that contain information of interest to an investigator.

Examining the plist files will reveal the nearby attractions depicted above. Information in this location includes Facebook user IDs, Facebook friends, etc.

Similar to the Maps folder, files containing geo-coordinates can often be encrypted. If you export these files for review within the Property List Viewer, you will notice that the content is encrypted. Viewing these files from within BlackLight's built-in Strings viewer provides access to the data within the embedded plist file.

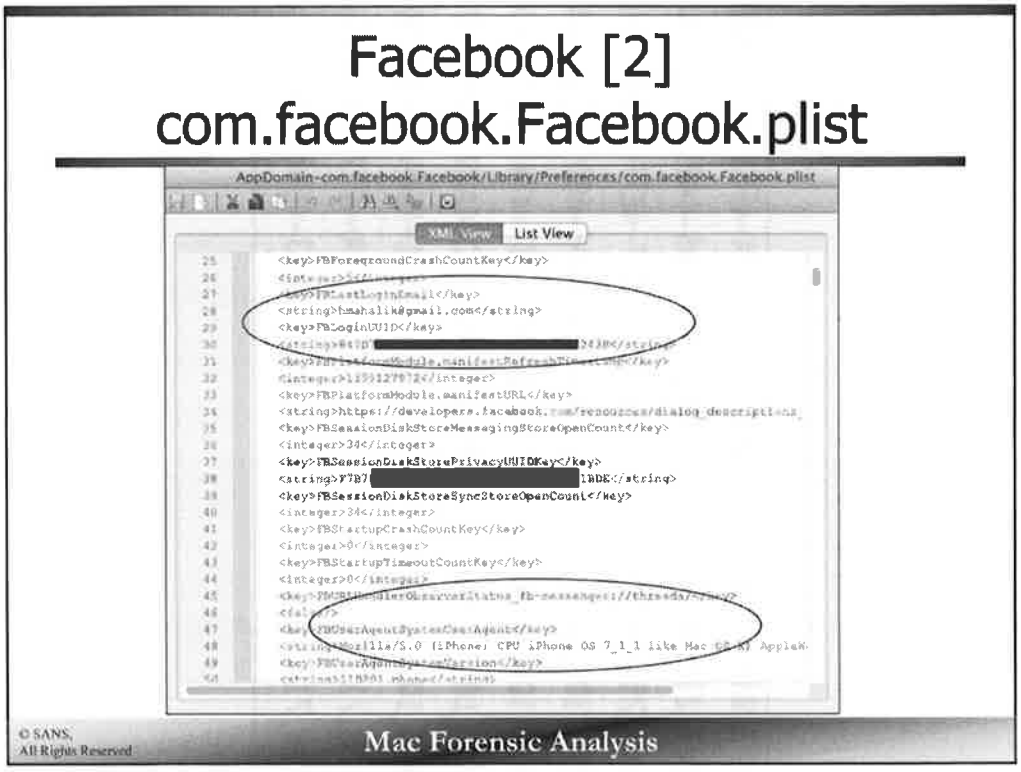

com.facebook.Facebook 2013-12-29 (UTC)
Library 2013-12-29 (UTC)
Cookies 2013-12-29 (UTC)
Cookies.binarycookies 2013-12-29 (UTC)
Preferences 2013-12-29 (UTC)
1097829115.plist 2013-12-29 (UTC)
1097829115.session.plist 2013-12-29 (UTC)
com.apple.mobileslideshow.plist 2013-02-04 (UTC)
com.apple.youtube.dp.plist 2013-12-08 (UTC)
com.apple.youtubeframework.plist 2012-06-20 (UTC)
TCity"B
FBNearbyPlace
FBNearbyPlace
FBNearbyDataSet_
FBNearbyDataSet
-./01
U @#
MNOPQRSTUVW
Nationals Park_
151128858232318
WNS.base
NS.relative
Zhttps://fbcdn-profile-a.akamaihd.net/hprofile-ak-ash4/203605_151128858232318_4837662_n.jpg
Embedded plist content can be manually carved or viewed within BlackLight "Strings"
URL points to image of Nationals Park viewed within Facebook application



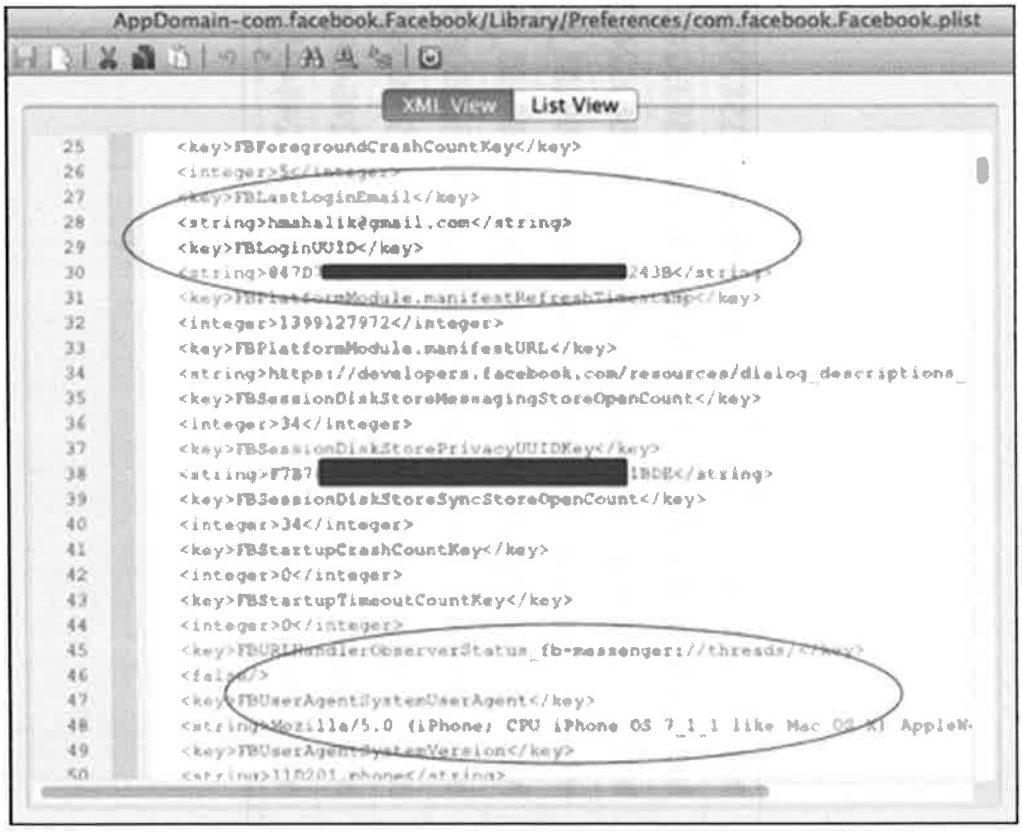
The com.facebook.Facebook.plist file contains user information as well as images from the Facebook application that have been cached on the device. Images are stored as a URL string which points to an associated JPG image.

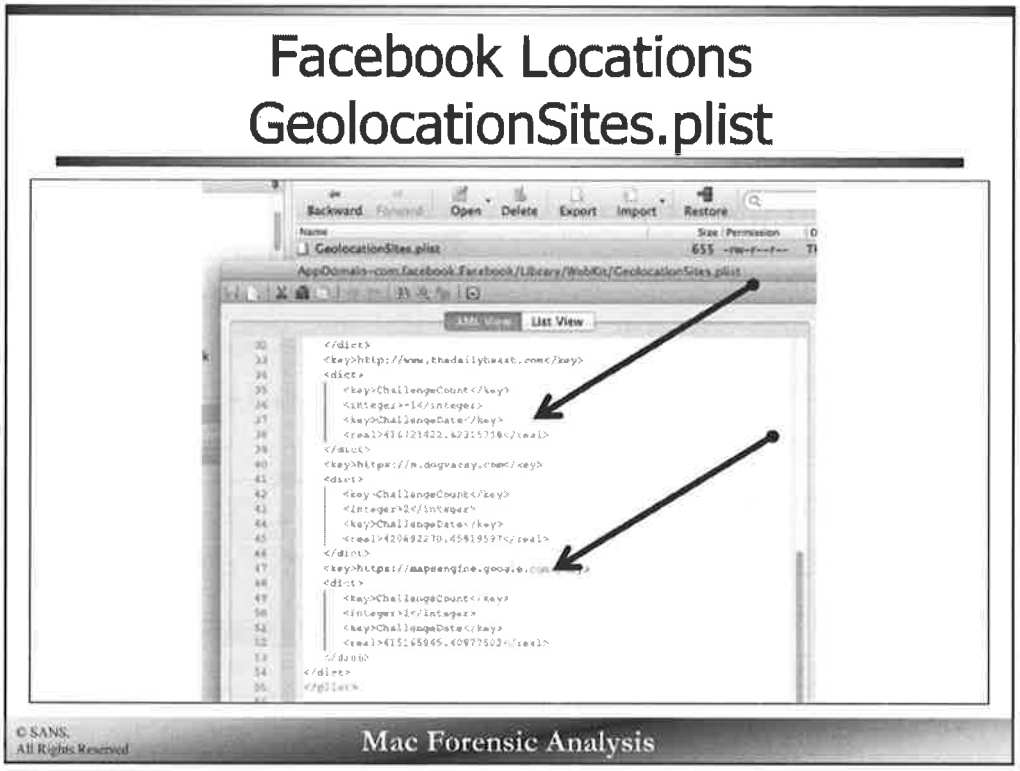
▼FBUserInfo	Dictionary	(1 item)
▼1097829115	Dictionary	(1 item)
▼1097829115	Dictionary	(7 items)
af	Boolean	☐
betaBuildAuthorized	Boolean	☐
cs	Boolean	☑
name	String	Lee Crognale
pic	String	https://fbcdn-profile-a.akama
pic_square	String	https://fbcdn-profile-a.akama

http://photos-d.ak.fbcdn.net/hphotos-ak-ash3/558700_4168839374772_844	String	30E055D7-4684-4F63-B1B8-4D371F9
http://photos-d.ak.fbcdn.net/hphotos-ak-ash3/563324_3949569453161_255	String	55D0D3D7-E1B5-44B8-B61C-BA44DA
http://photos-d.ak.fbcdn.net/hphotos-ak-ash3/563324_3949569453161_255	String	2FEBA782-DCFF-4E7F-B208-0885919
http://photos-d.ak.fbcdn.net/hphotos-ak-prn1/69183_4331862370245_2109	String	D80A44D5-3CAF-4603-899C-0488D8
http://photos-d.ak.fbcdn.net/hphotos-ak-prn1/69183_4331862370245_2109	String	D5952ADC-8187-42BE-B17D-39D2F0
http://photos-e.ak.fbcdn.net/hphotos-ak-ash3/580006_3823466420664_101	String	B535FB27-0A5C-4A2B-BCAB-F8F790S
http://photos-e.ak.fbcdn.net/hphotos-ak-ash3/580006_3823466420664_101	String	3EE48F33-A9C1-46B9-BA4B-1099E22
http://photos-e.ak.fbcdn.net/hphotos-ak-ash4/246491_4394431294429_719	String	66841E0D-763A-4E27-83C7-803D65
http://photos-e.ak.fbcdn.net/hphotos-ak-ash4/246491_4394431294429_719	String	807C5A7E-3ED3-4694-9CFD-118522

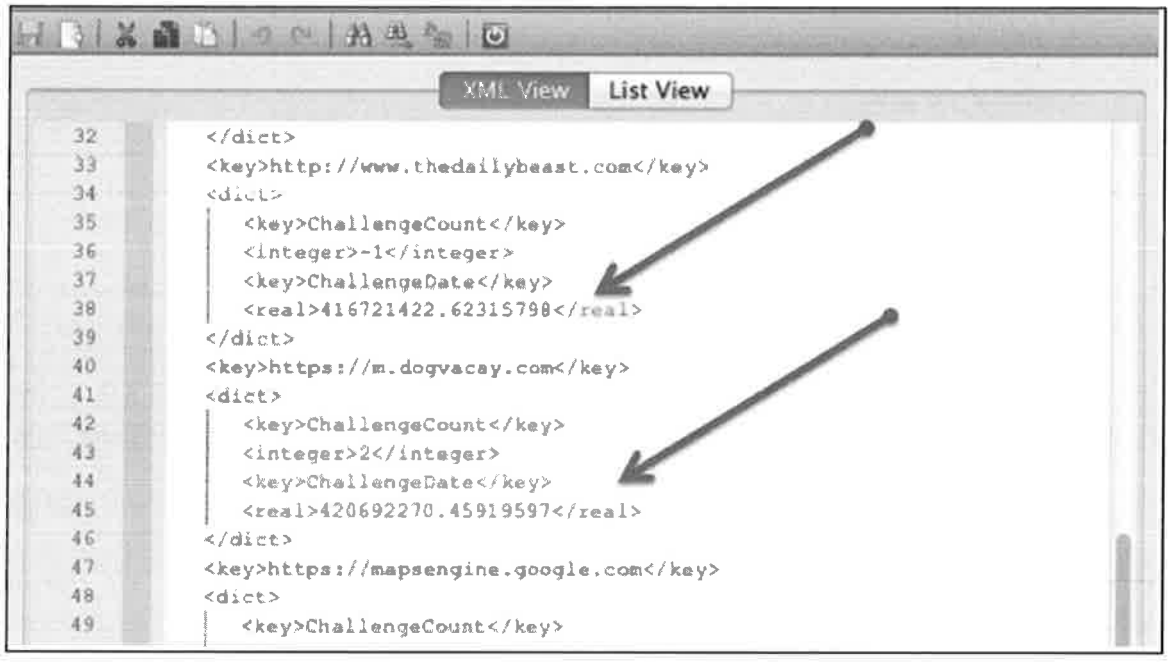


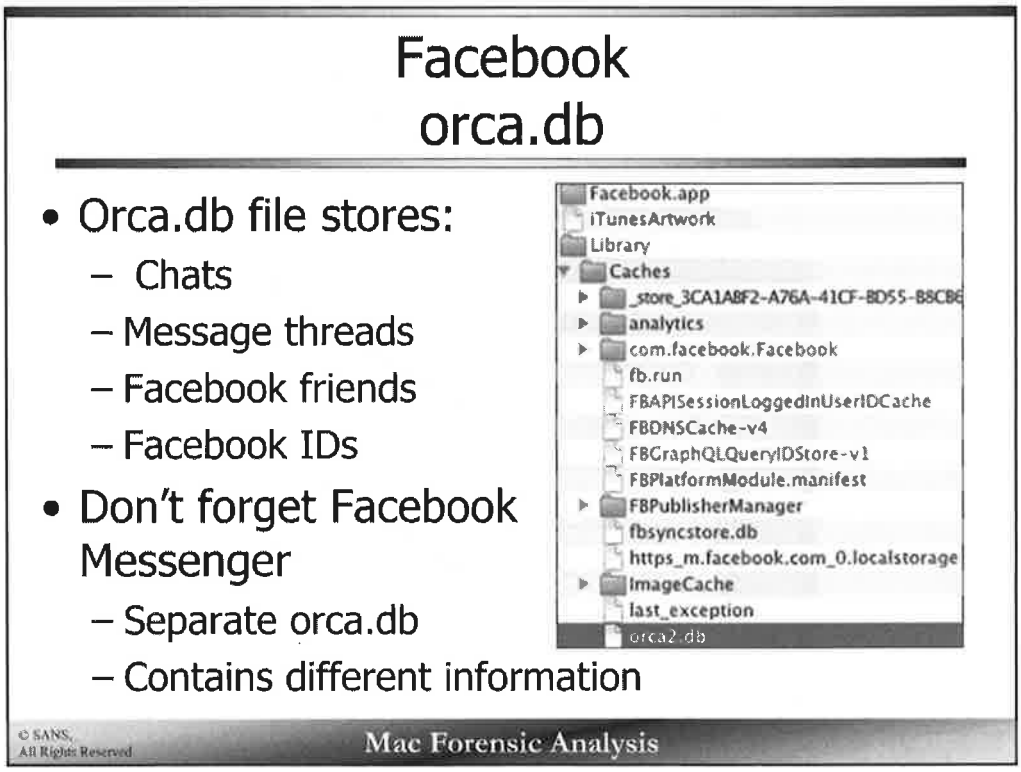
Facebook data pulled from an iPhone 5S backup files is shown below. The com.facebook.Facebook.plist file contains user information such as account login information, the UUID for the account, device information and more.





Location information can be pulled from backup files even when the Facebook application folder is not captured or parsed in the backup file. Below, we can see location information associated to two websites. These websites appeared as advertisements on the users device. When selected, the application captures the users location and allows for them to search locations of their choice. For example, the site https://m.dogvacy.com was selected and areas on Northern VA were searched for petcare while the user is on vacation. Location information such as this may be relevant to your investigation. This data was recovered from GeolocationSites.plist file using iBackup Bot.

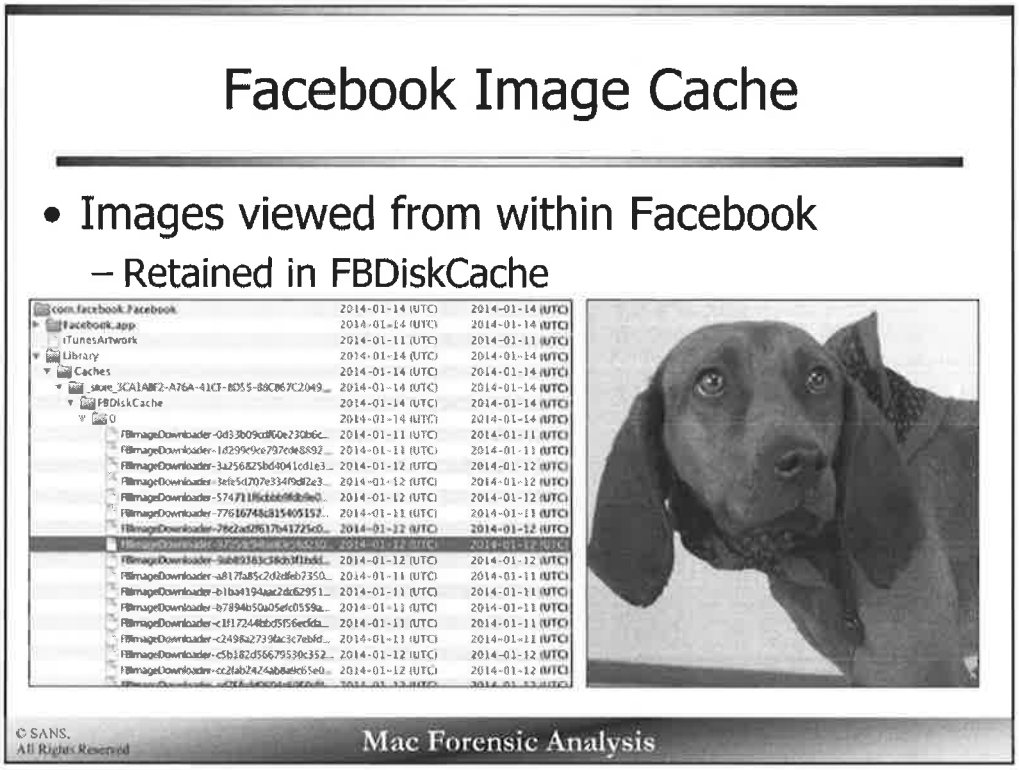




The orca.db file is a SQLite database file that contains the bulk of the Facebook application messaging content. Both Facebook and Facebook Messenger will have their own orca.db files which can contain completely different information.

Orca.db tables include:

- store_version
- app_state
- threads
- messages
- users
- profile_pic_urls
- idents
- members
- sqlite_stat1



Images recovered from com.facebook.Facebook/Facebook.app/Library/Cahces/_store_XXX/FBDiskCache/0/ are images that were viewed while utilizing the Facebook application. These temporary images are downloaded to the device.

com.facebook.Facebook	2014-01-14 (UTC)	2014-01-14 (UTC)
Facebook.app	2014-01-14 (UTC)	2014-01-14 (UTC)
iTunesArtwork	2014-01-11 (UTC)	2014-01-11 (UTC)
Library	2014-01-14 (UTC)	2014-01-14 (UTC)
Caches	2014-01-14 (UTC)	2014-01-14 (UTC)
_store_3CA1A8F2-A76A-41CF-8D55-88CB67C2049...	2014-01-14 (UTC)	2014-01-14 (UTC)
FBDiskCache	2014-01-14 (UTC)	2014-01-14 (UTC)
0	2014-01-14 (UTC)	2014-01-14 (UTC)
FBImageDownloader-0d33b09cdf60e230b6c...	2014-01-11 (UTC)	2014-01-11 (UTC)
FBImageDownloader-1d299c9ce797cde8892...	2014-01-11 (UTC)	2014-01-11 (UTC)
FBImageDownloader-3a256825bd4041cd1e3...	2014-01-12 (UTC)	2014-01-12 (UTC)
FBImageDownloader-3efe5d707e334f9df2e3...	2014-01-12 (UTC)	2014-01-12 (UTC)
FBImageDownloader-5747118fcbbb9fdb9e0...	2014-01-12 (UTC)	2014-01-12 (UTC)
FBImageDownloader-77616748c815405152...	2014-01-11 (UTC)	2014-01-11 (UTC)
FBImageDownloader-78c2ad2f617b41725c0...	2014-01-12 (UTC)	2014-01-12 (UTC)
FBImageDownloader-9705dc948a80e58d230...	2014-01-12 (UTC)	2014-01-12 (UTC)
FBImageDownloader-9ab89383c38cb3f1bdd...	2014-01-12 (UTC)	2014-01-12 (UTC)
FBImageDownloader-a817fa85c2d2dfeb7350...	2014-01-11 (UTC)	2014-01-11 (UTC)
FBImageDownloader-b1ba4394aac2dc629511...	2014-01-11 (UTC)	2014-01-11 (UTC)
FBImageDownloader-b7894b50a05efc0559a...	2014-01-11 (UTC)	2014-01-11 (UTC)
FBImageDownloader-c1f17244bbd5f56edda...	2014-01-11 (UTC)	2014-01-11 (UTC)
FBImageDownloader-c2498a2739fac3c7ebfd1...	2014-01-11 (UTC)	2014-01-11 (UTC)
FBImageDownloader-c5b182d56679530c352...	2014-01-12 (UTC)	2014-01-12 (UTC)
FBImageDownloader-cc2fab2424ab8a9d65e0...	2014-01-12 (UTC)	2014-01-12 (UTC)

LinkedIn [1]

- LOTS of user data on the device
- Mobile/Applications/LinkedIn/Documents
  - Notifications_data_center_key

nId	String	0:MBR_18709939:28
nType	String	ACCEPTED_YOUR_CONNECTION_REQUEST
resourcePath	String	/li/v1/people/3612169/profile?authToken=name:rJA
seen	Boolean	True
▼ socialHeader	Dictionary	(6 items)
headerText	String	Jessic
pictureLogo	String	person
pictureUrl	String	http://media.linkedin.com/mpr/mpr/p/4/005/027/1
▼ socialSummary	Dictionary	(1 item)
timestamp	Number	1396819627535
text1	String	accepted your invitation to connect
tType	String	sht5
timestamp	Number	1396819627535

© SANS.
All Rights Reserved

Notifications_data_center_key – notification status such as endorsements and friend request acceptances

Ia_Data – suggestions, profile information

connectionsDiskCacheKey – friends who made recent connections

LinkedIn.sqlite

NUSSUBTEMPLATE – contains the friend feed

nId	String	0:MBR_18709939:28
nType	String	ACCEPTED_YOUR_CONNECTION_REQUEST
resourcePath	String	/li/v1/people/3612169/profile?authToken=name:rJA
seen	Boolean	True
▼ socialHeader	Dictionary	(6 items)
headerText	String	Jessic
pictureLogo	String	person
pictureUrl	String	http://media.linkedin.com/mpr/mpr/p/4/005/027/1
▼ socialSummary	Dictionary	(1 item)
timestamp	Number	1396819627535
text1	String	accepted your invitation to connect
tType	String	sht5
timestamp	Number	1396819627535

LinkedIn [2]

- Mobile/Applications/LinkedIn/Documents
  - SearchCache.sqlite

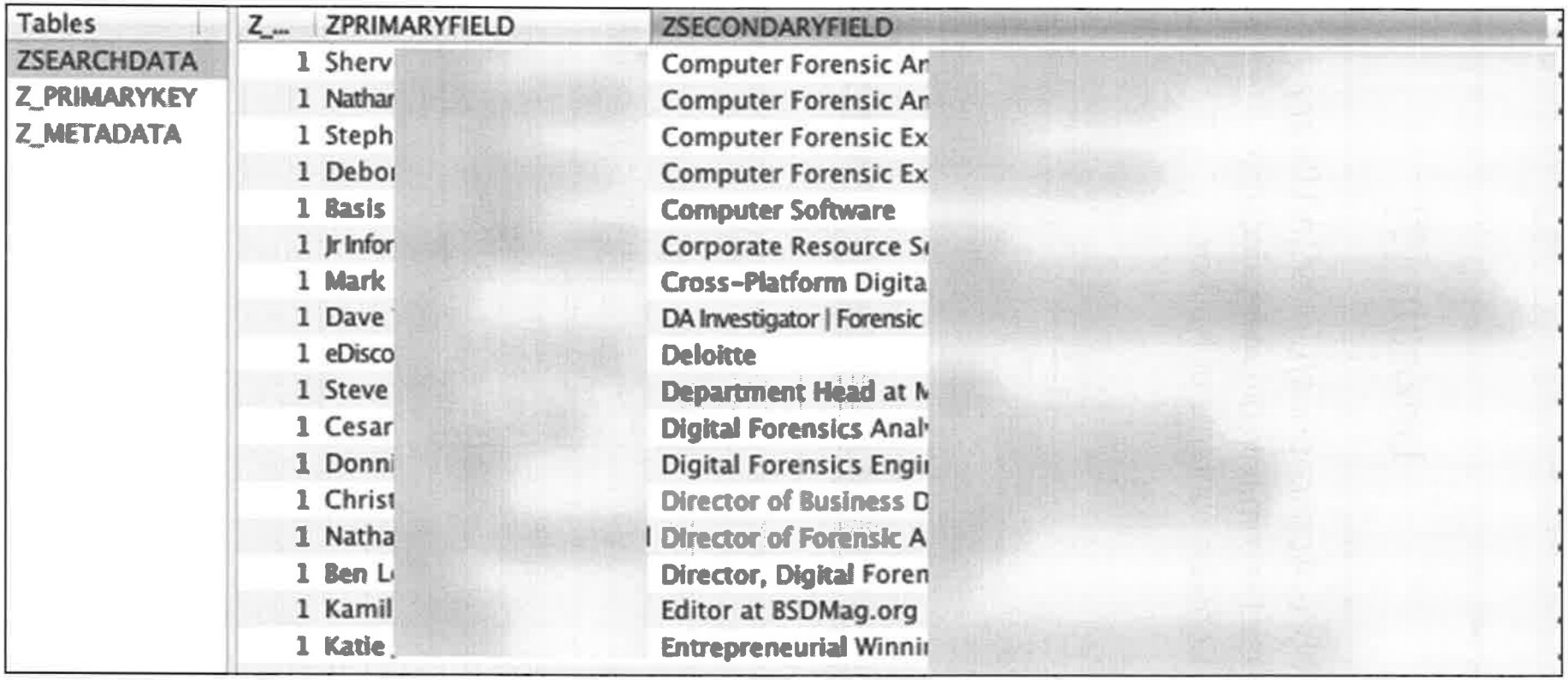
Tables	Z_	ZPRIMARYFIELD	ZSECONDARYFIELD
ZSEARCHDATA	1	Sherv	Computer Forensic Ar
Z_PRIMARYKEY	1	Nathar	Computer Forensic Ar
Z_METADATA	1	Steph	Computer Forensic Ex
	1	Debor	Computer Forensic Ex
	1	Basis	Computer Software
	1	Jr Infor	Corporate Resource Sc
	1	Mark	Cross-Platform Digita
	1	Dave	DA Investigator I Forensic
	1	eDisco	Deloitte
	1	Steve	Department Head at M
	1	Cesar	Digital Forensics Anal
	1	Donn	Digital Forensics Engi
	1	Christ	Director of Business D
	1	Natha	Director of Forensic A
	1	Ben L	Director, Digital Foren
	1	Kamil	Editor at BSDMag.org
	1	Katie	Entrepreneurial Winni

© SANS,
All Rights Reserved

This database provides access to each LinkedIn connection.

SearchCache.sqlite

ZSEARCHDATA table contains friends lists name and job description



Tables	Z_...	ZPRIMARYFIELD	ZSECONDARYFIELD
ZSEARCHDATA	1	Sherv	Computer Forensic Ar
Z_PRIMARYKEY	1	Nathar	Computer Forensic Ar
Z_METADATA	1	Steph	Computer Forensic Ex
	1	Debor	Computer Forensic Ex
	1	Basis	Computer Software
	1	Jr Infor	Corporate Resource S
	1	Mark	Cross-Platform Digita
	1	Dave	DA Investigator | Forensic
	1	eDisco	Deloitte
	1	Steve	Department Head at M
	1	Cesar	Digital Forensics Anal
	1	Donni	Digital Forensics Engi
	1	Christ	Director of Business D
	1	Natha	Director of Forensic A
	1	Ben L	Director, Digital Foren
	1	Kamil	Editor at BSDMag.org
	1	Katie	Entrepreneurial Winni

LinkedIn [3]

- Mobile/Applications/LinkedIn/Library
  - Cachescom.linkedin.messagecache
  - Contains contents of Mailbox

Key	Type	Value
▼ Item 46	Dictionary	(2 items)
$class	UID	27
NS.string	String	I'd like to add you to my professional network on LinkedIn.
▼ Item 47	Dictionary	(2 items)
$class	UID	27
NS.string	String	Inbox
Item 48	Number	1397224202827
Item 49	Boolean	False

© SANS.
All Rights Reserved

Key	Type	Value
▼ Item 46	Dictionary	(2 items)
$class	UID	27
NS.string	String	I'd like to add you to my professional network on LinkedIn.
▼ Item 47	Dictionary	(2 items)
$class	UID	27
NS.string	String	Inbox
Item 48	Number	1397224202827
Item 49	Boolean	False

LinkedIn [4]

- Binary cookies stored per application

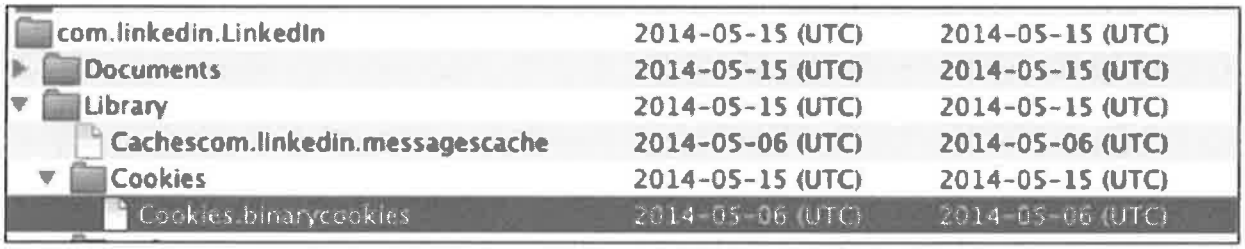
- Binary Cookie format is the same as Safari iOS Cookies

```
Awww.linkedin.comvisit/"v=1&M"
A.touch.www.linkedin.comlim_abi/1399391759659:truet
A.touch.www.linkedin.comlim_abi_splash/1399391759659:true
A.touch.www.linkedin.comlim_ut/2dc9a315-2ec2-494f-aab2-a40374832528
A.touch.www.linkedin.comlim_sr/li/v2/streamrelevance:1399391753497
```

© SANS, All Rights Reserved

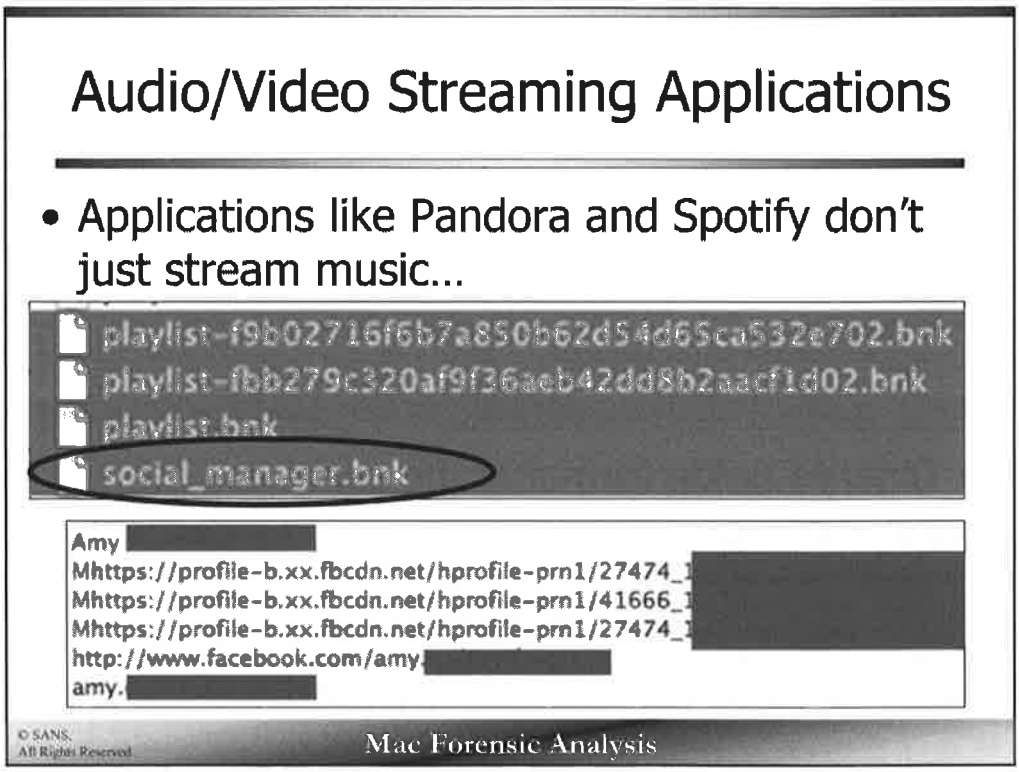
Some applications store the own Cookie data for the same reason that Safari stores cookies. This data can be stored for each application independently.

- Binary cookies stored per application



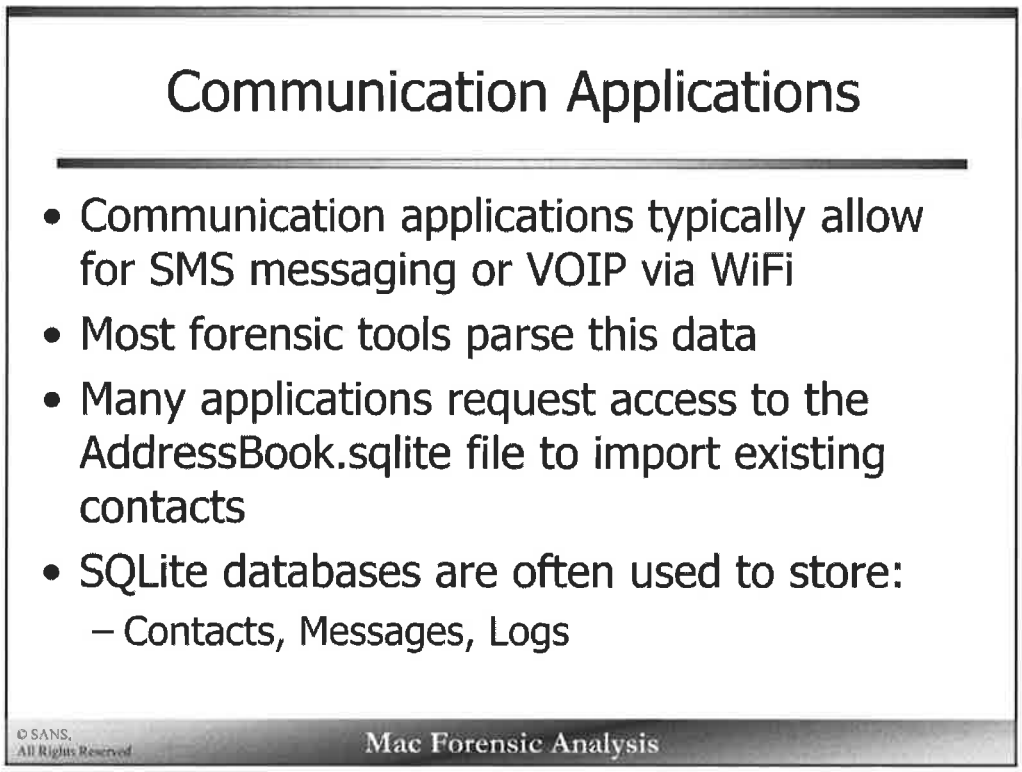
- Binary Cookie format is the same as Safari iOS Cookies

```
Awww.linkedin.comvisit/"v=1&M"
A.touch.www.linkedin.comlim_abi/1399391759659:truet
A.touch.www.linkedin.comlim_abi_splash/1399391759659:true
A.touch.www.linkedin.comlim_ut/2dc9a315-2ec2-494f-aab2-a40374832528
A.touch.www.linkedin.comlim_sr/li/v2/streamrelevance:1399391753497
```



Audio and Video applications don't just store audio/video files. These applications have become more akin to Social Networking applications in that they share users, playlists, and geo-coordinates. In addition to the information you would expect to find in audio and video streaming applications, examine individual files for data that is shared between applications. Spotify users who share content within the Facebook application can have profile information listed within the Spotify application.

In the example above, social_manager.bnk will contain a list of a user's Facebook connections who are also using Spotify.



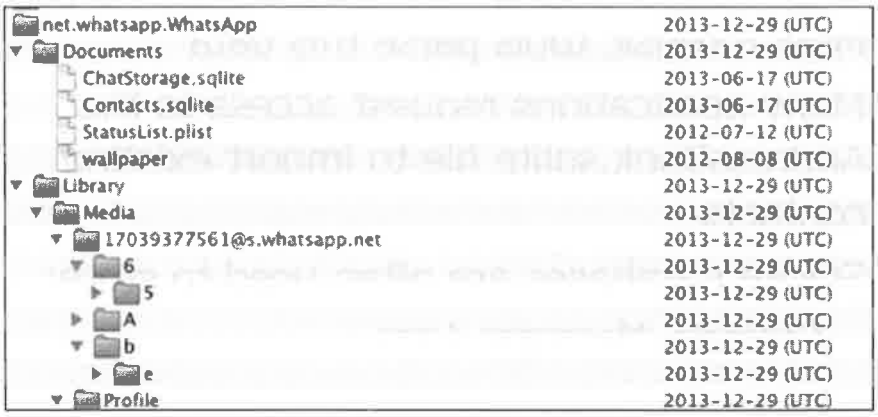
Some common messaging applications include:

- WhatsApp
- Viber
- Tango
- Nimbuzz
- HeyWire
- SnapChat
- Line
- Kik
- Grouptime
- Groupme
- Voxer
- DingDong
- Wickr
- Silent Circle

Some features that are common to most of these applications include address lists, messages, plist files containing user data, Documents (audio/video/image files).

WhatsApp [1]

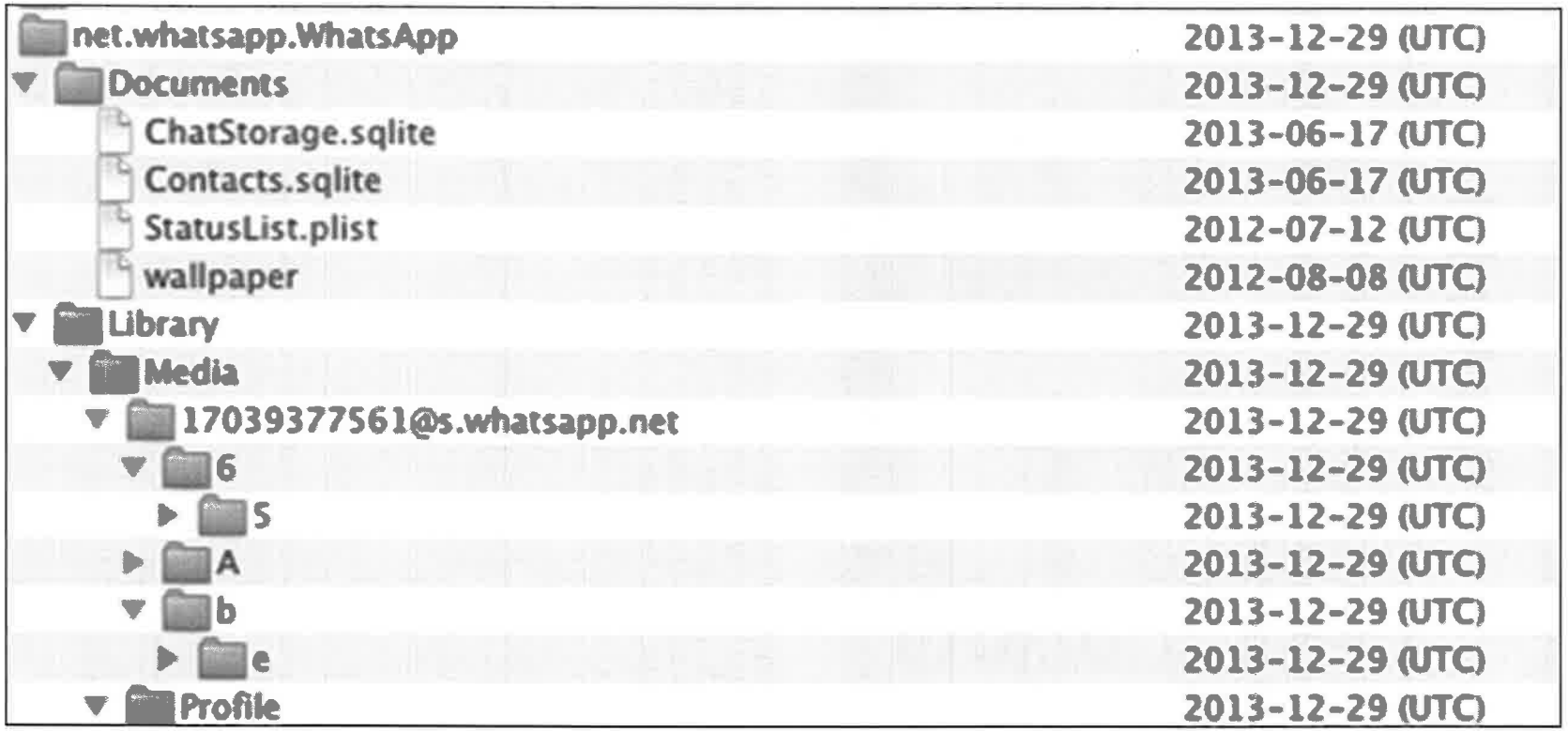
- Much of the communication data is retained within subfolders of the application



© SANS, All Rights Reserved

WhatsApp is one of the most widely used messaging applications for iDevices. Fortunately, this application stores much of its information within the application sandbox. The bulk of the messaging data will be recovered from the ZWAMESSAGE Table within the ChatStorage.sqlite database file.

Corresponding media attachments will be located within the Media directory and grouped by User Number (ex: 17039377561@s.whatsapp.net will contain the media files shared with this user).


net.whatsapp.WhatsApp 2013-12-29 (UTC)
Documents 2013-12-29 (UTC)
ChatStorage.sqlite 2013-06-17 (UTC)
Contacts.sqlite 2013-06-17 (UTC)
StatusList.plist 2012-07-12 (UTC)
wallpaper 2012-08-08 (UTC)
Library 2013-12-29 (UTC)
Media 2013-12-29 (UTC)
17039377561@s.whatsapp.net 2013-12-29 (UTC)
6 2013-12-29 (UTC)
5 2013-12-29 (UTC)
A 2013-12-29 (UTC)
b 2013-12-29 (UTC)
e 2013-12-29 (UTC)
Profile 2013-12-29 (UTC)

WhatsApp [2]

- Many communication/networking applications will request access to the iOS Address Book
- Review TCC.db to confirm accesses

ZFIRSTNAME	ZFULLNAME
Jim	Jim Par
Laura	Laura I
Christia	Christi
Trudy	Trudy
Kathryn	Kathry
Christia	Christi
Kelly	Kelly C
spector	specto
Chris	Chris S

© SANS, All Rights Reserved

Upon first look, it may appear that every username contained within the WhatsApp Contacts.sqlite file have been communicating using WhatsApp. It is common for many messaging applications to request access to the Native iOS AddressBook.sqlite file to import contacts. Review the ChatStorage.sqlite database file to confirm communication between individuals appearing in the Contacts.sqlite database. Access the TCC.db file and review the relevant logs for WhatsApp to see if it has been granted access to the kTCCServiceAddressBook file.

WhatsApp [3]

- ChatStorage.sqlite

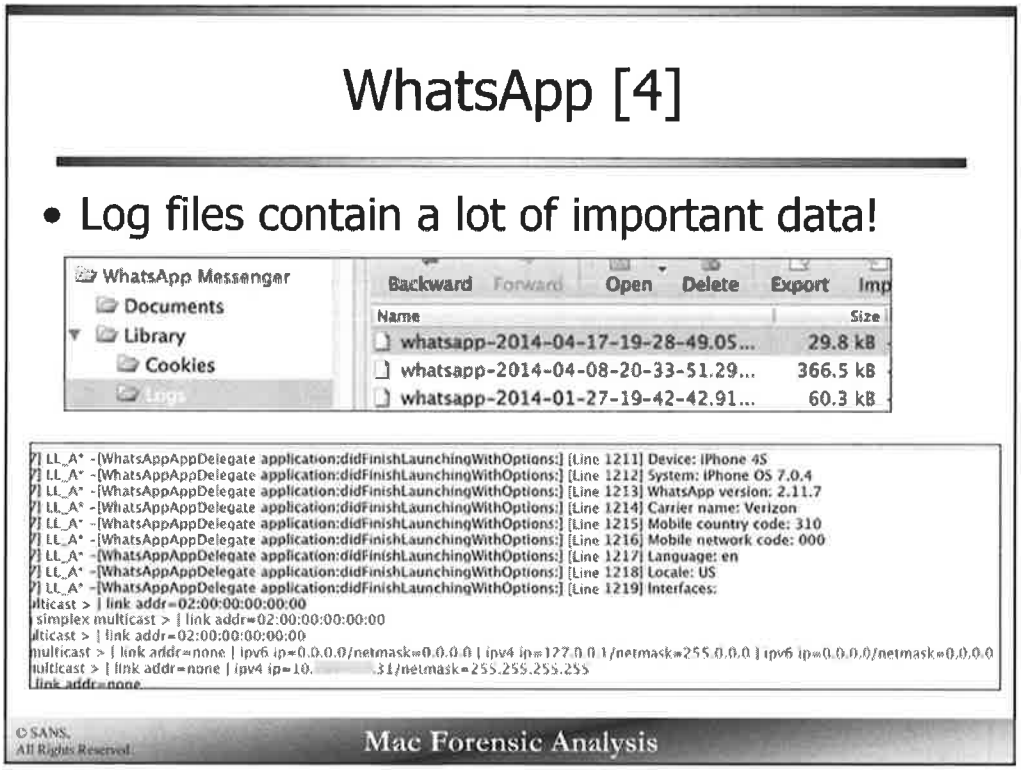
175	@s.whatsapp.net	Vic	1397004057-47	Man shoe
			1397003647-151	Haha
175	@s.whatsapp.net	Vic	1397004057-64	Just got message from you on my phone that is invi
			1397003647-160	I just went in to change it and saw that pic in my ph
			1397003647-167	Changed my PCI
			1397003647-173	Pic
			1397003647-180	Did it change?
175	@s.whatsapp.net	Vic	1397004057-88	You are using your regular phone number, right?
			1397003647-195	It's probably bc I sign into multiple phones with my
175	@s.whatsapp.net	Vic	1397004057-99	Yeah, you are Calvin now
			1397003647-207	It's weird that it showed him in my profile
			1397003647-214	Now that shoe is in my camera reel
175	@s.whatsapp.net	Vic	1397004057-118	Is mine a pic of bash and max?
			1397003647-226	But I have logged into a lot of diff phonemes
			1397003647-238	Yeah
			1397003647-232	Phones
175	@s.whatsapp.net	Vic	1397004057-133	And you use your phone number for all of them?
175	@s.whatsapp.net	Vic	1397004057-141	I wanted to get assigned a new number but I don't k

© SANS,
All Rights Reserved

ChatStorage.sqlite

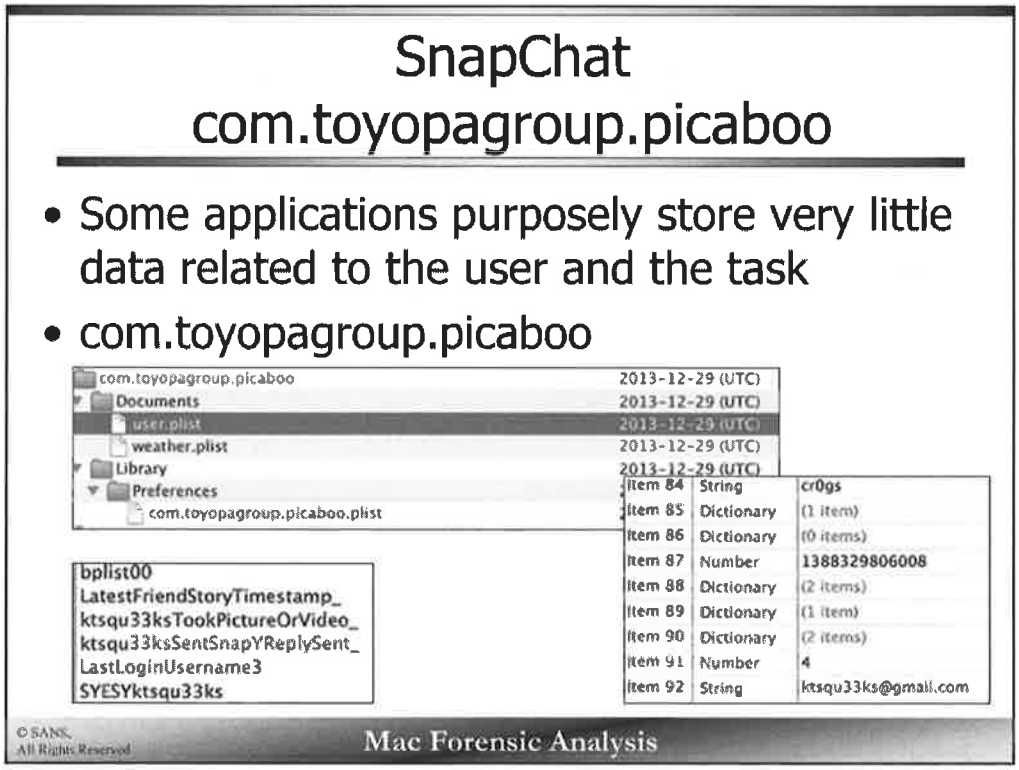
ZWACHATSESSION

175	@s.whatsapp.net	Vic	1397004057-47	Man shoe
			1397003647-151	Haha
175	@s.whatsapp.net	Vic	1397004057-64	Just got message from you on my phone that is invit
			1397003647-160	I just went in to change it and saw that pic In my ph
			1397003647-167	Changed my PCI
			1397003647-173	Pic
			1397003647-180	Did it change?
175	@s.whatsapp.net	Vic	1397004057-88	You are using your regular phone number, right?
			1397003647-195	It's probably bc I sign into multiple phones with my
175	@s.whatsapp.net	Vic	1397004057-99	Yeah, you are Calvin now
			1397003647-207	It's weird that it showed him in my profile
			1397003647-214	Now that shoe is in my camera reel
175	@s.whatsapp.net	Vic	1397004057-118	Is mine a pic of bash and max?
			1397003647-226	But I have logged into a lot of diff phonemes
			1397003647-238	Yeah
			1397003647-232	Phones
175	@s.whatsapp.net	Vic	1397004057-133	And you use your phone number for all of them?
175	@s.whatsapp.net	Vic	1397004057-141	I wanted to get assigned a new number but I don't k

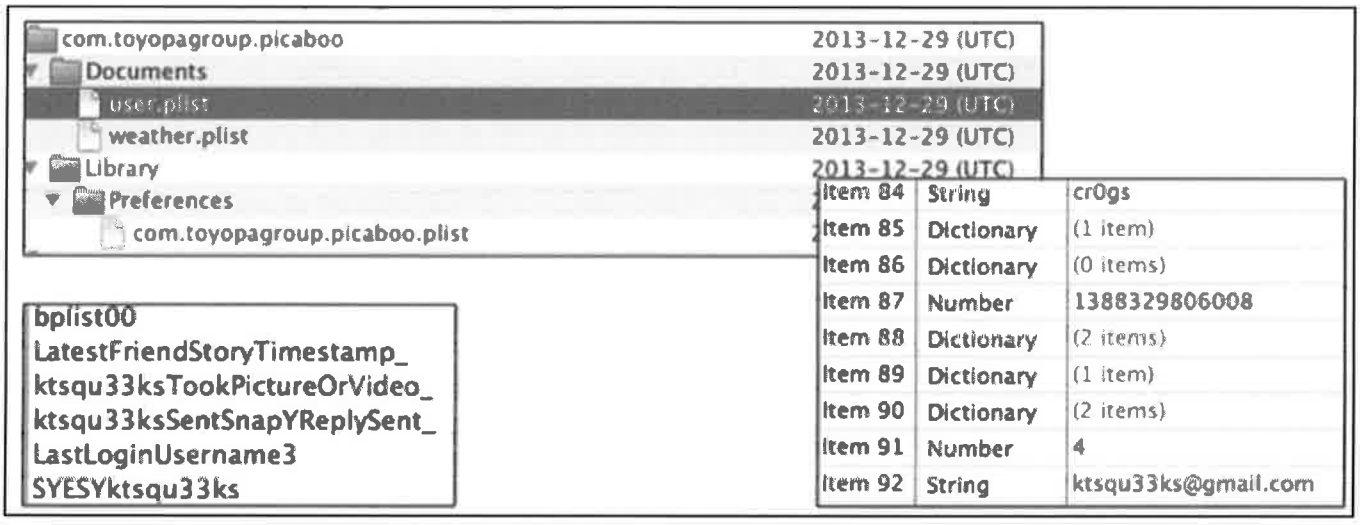


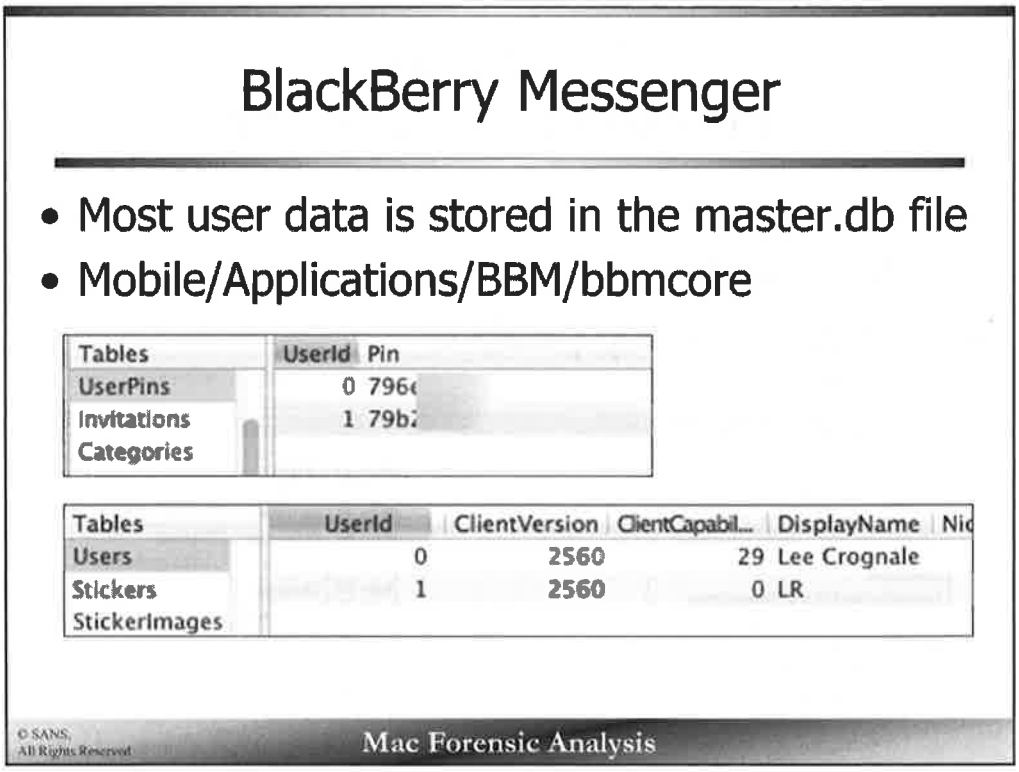
Review logs for ip addresses, device information, device phone number, WhatsApp messaging phone numbers, etc. This log file stores great information if you don't have access to the database file.


```
7] LL_A* -[WhatsAppAppDelegate application:didFinishLaunchingWithOptions:] [Line 1211] Device: iPhone 4S
7] LL_A* -[WhatsAppAppDelegate application:didFinishLaunchingWithOptions:] [Line 1212] System: iPhone OS 7.0.4
7] LL_A* -[WhatsAppAppDelegate application:didFinishLaunchingWithOptions:] [Line 1213] WhatsApp version: 2.11.7
7] LL_A* -[WhatsAppAppDelegate application:didFinishLaunchingWithOptions:] [Line 1214] Carrier name: Verizon
7] LL_A* -[WhatsAppAppDelegate application:didFinishLaunchingWithOptions:] [Line 1215] Mobile country code: 310
7] LL_A* -[WhatsAppAppDelegate application:didFinishLaunchingWithOptions:] [Line 1216] Mobile network code: 000
7] LL_A* -[WhatsAppAppDelegate application:didFinishLaunchingWithOptions:] [Line 1217] Language: en
7] LL_A* -[WhatsAppAppDelegate application:didFinishLaunchingWithOptions:] [Line 1218] Locale: US
7] LL_A* -[WhatsAppAppDelegate application:didFinishLaunchingWithOptions:] [Line 1219] Interfaces:
lticast > | link addr=02:00:00:00:00:00
 simplex multicast > | link addr=02:00:00:00:00:00
lticast > | link addr=02:00:00:00:00:00
multicast > | link addr=none | ipv6 ip=0.0.0.0/netmask=0.0.0.0 | ipv4 ip=127.0.0.1/netmask=255.0.0.0 | ipv6 ip=0.0.0.0/netmask=0.0.0.0
nulticast > | link addr=none | ipv4 ip=10.[redacted].31/netmask=255.255.255.255
 link addr=none
```



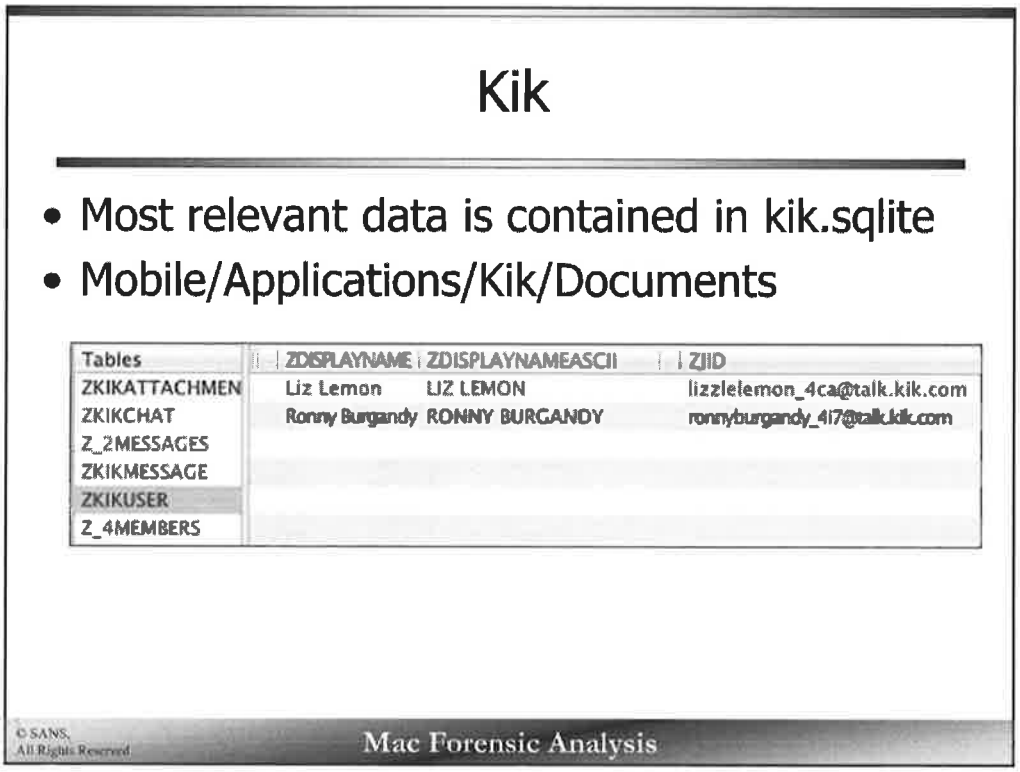
Not all applications will retain important user data. Some applications, like SnapChat are designed to save almost zero content, most likely the result of increasing mobile forensic investigations. Very limited information is stored by the SnapChat application. The user.plist file contains references to other SnapChat users who communicate with the profile set up by "ktsqu33ks@gmail.com", but no images or chats related to their communication were saved.





Tables of particular interest include:

- Contacts
- Conversations
- Invitations
- Locations
- Profile
- Text Messages
- User Pins
- Users



Relevant user data is contained in the following tables:

ZKIKATTACHMENT

ZKIKCHAT

ZKIKMESSAGE

ZKIKUSER

The Kik application installs a lot of miscellaneous data to the device upon install.

Nimbuzz [1]

- Account information to include username, last login and more are contained in plist files
  - 'Username.'plist
  - Com.nimbuzz.plist

	Type	Value
	String	iPhone OS, 7.1
	Dictionary	(2 items)
	Dictionary	(2 items)
	Dictionary	(2 items)
	String	02:00:00:00:00:00
	String	iPhone6,1
	Number	6188
	String	mylloydxmas@nimbuzz.com
	Number	0
	Number	63598354
	String	mylloydxmas@nimbuzz.com
	String	3.5.0
	Number	139[redacted]
	String	NULL
	Number	5864
	String	en

© SANS, All Rights Reserved

	Type	Value
	String	iPhone OS, 7.1
	Dictionary	(2 items)
	Dictionary	(2 items)
	Dictionary	(2 items)
	String	02:00:00:00:00:00
	String	iPhone6,1
	Number	6188
	String	mylloydxmas@nimbuzz.com
	Number	0
	Number	63598354
	String	mylloydxmas@nimbuzz.com
	String	3.5.0
	Number	139[redacted]
	String	NULL
	Number	5864
	String	en

Nimbuzz [2]

- Nimbuzz.db – username, PLAINTEXT passwords, phone number

Database
▼ Tables
account
account_settings
call_history

	id	username	password	last_login
1	1	mylloydxmas	[redacted]	1

Database
▼ Tables
account
account_settings
call_history
call_out_settings
chatItems
chat_history
chat_history_backup
client_configuration

	account_id	id	value
1	1	17	311
2	1	18	480
3	1	19	
4	1	14	1
5	1	15	20140507195254
6	1	5	
7	1	8	US
8	1	22	+157[redacted]

© SANS, All Rights Reserved

Database
▼ Tables
account
account_settings
call_history

	id	username	password	last_login
1	1	mylloydxmas	[redacted]	1

Database
▼ Tables
account
account_settings
call_history
call_out_settings
chatItems
chat_history
chat_history_backup
client_configuration

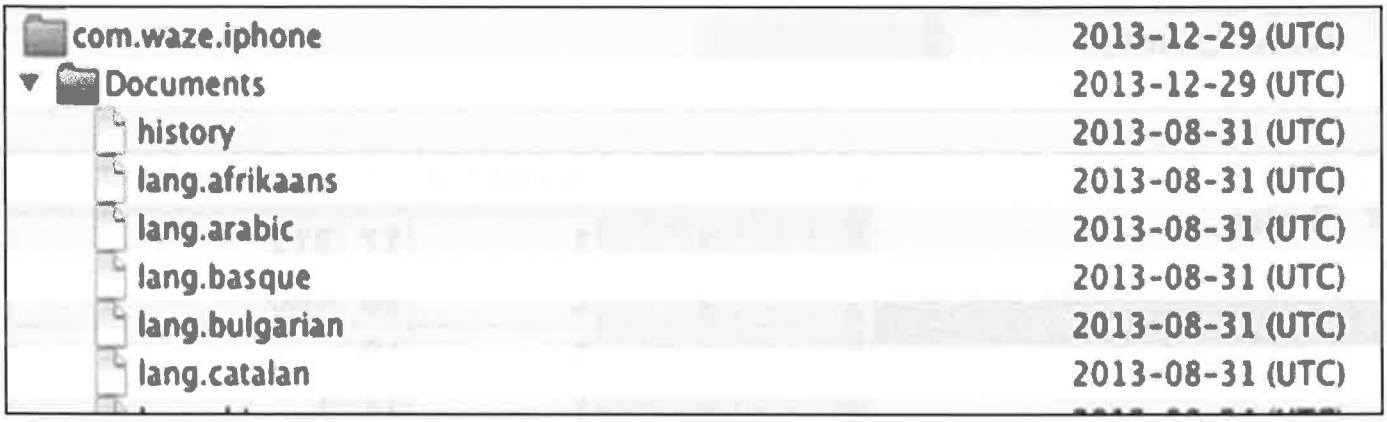
	account_id	id	value
1	1	17	311
2	1	18	480
3	1	19	
4	1	14	1
5	1	15	20140507195254
6	1	5	
7	1	8	US
8	1	22	+157[redacted]

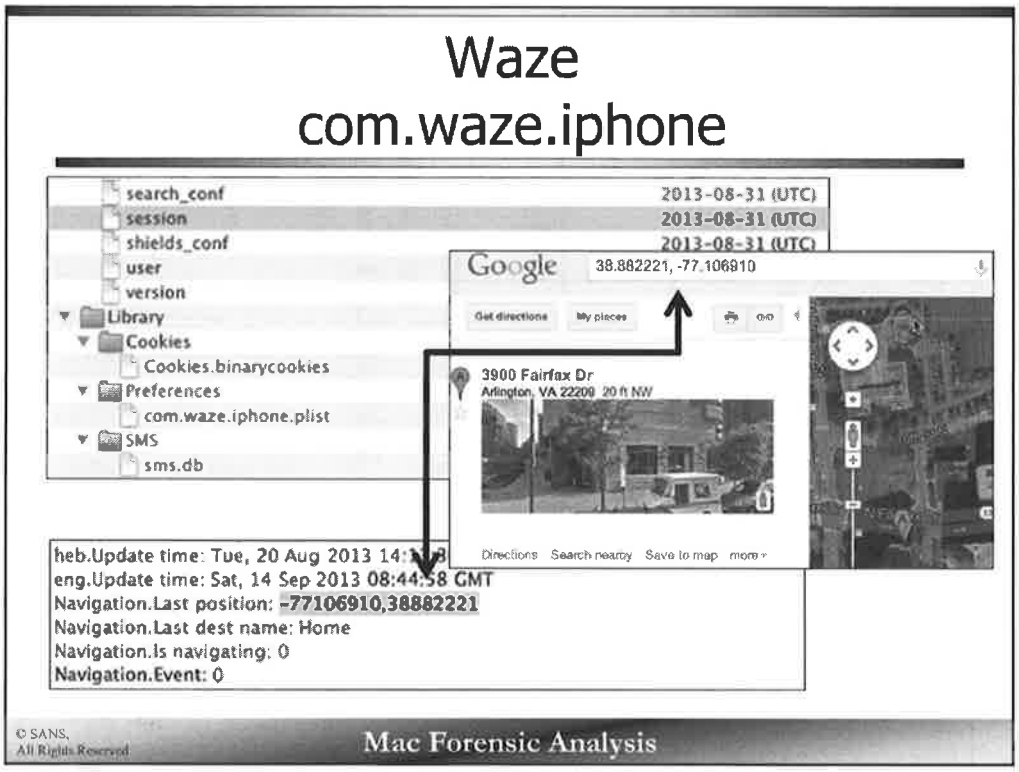
Mapping/Tracking Applications

- Applications like Waze allow for a more interactive driving experience
- In addition to mapped locations, look for contacts and chats associated with mapping and tracking apps

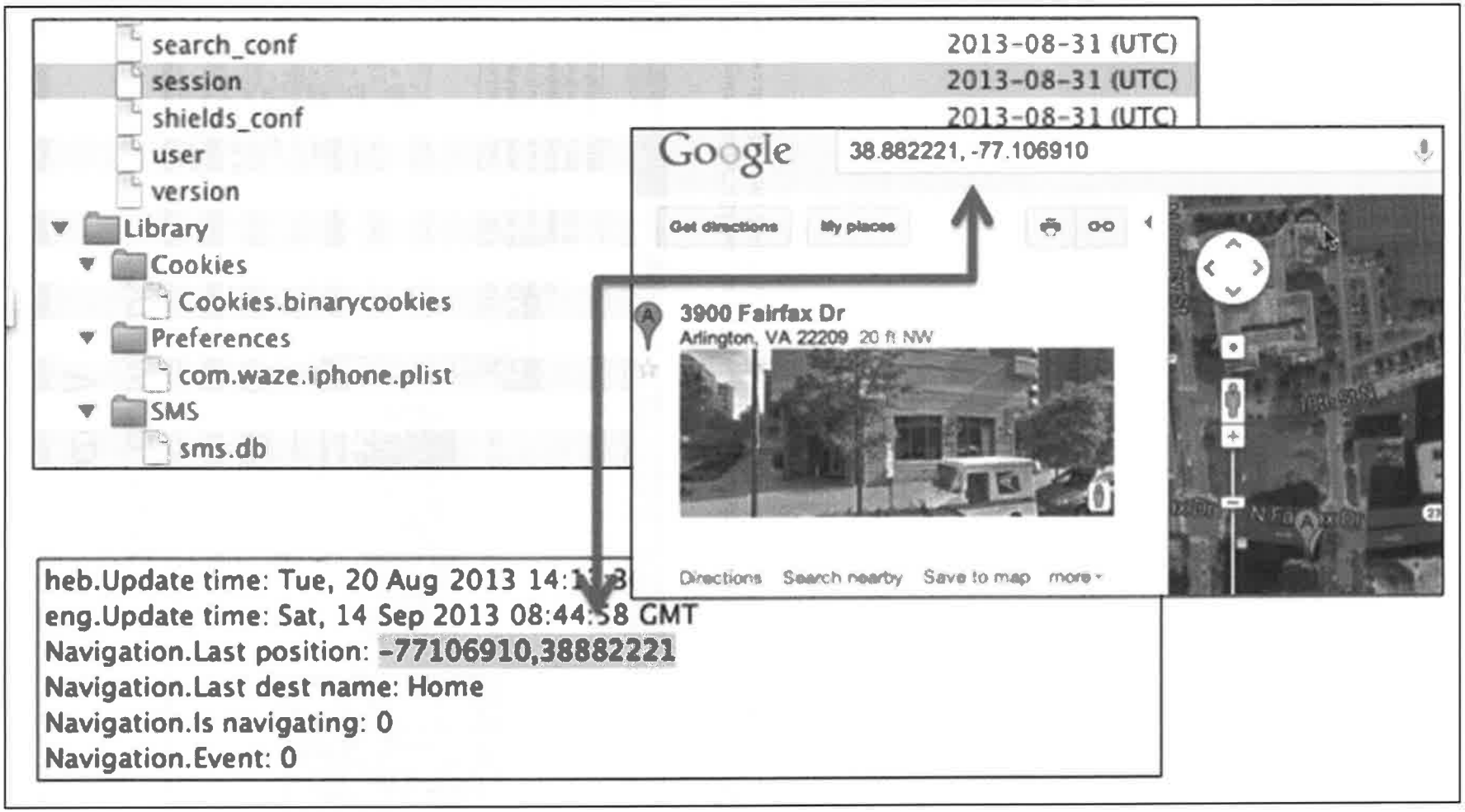
© SANS, All Rights Reserved

As previously mentioned, third-party applications are combining multiple features to make the application more robust. The navigation application, Waze incorporates mapping, chat messaging, check-ins, and more and all of this data is contained within the application's directory.

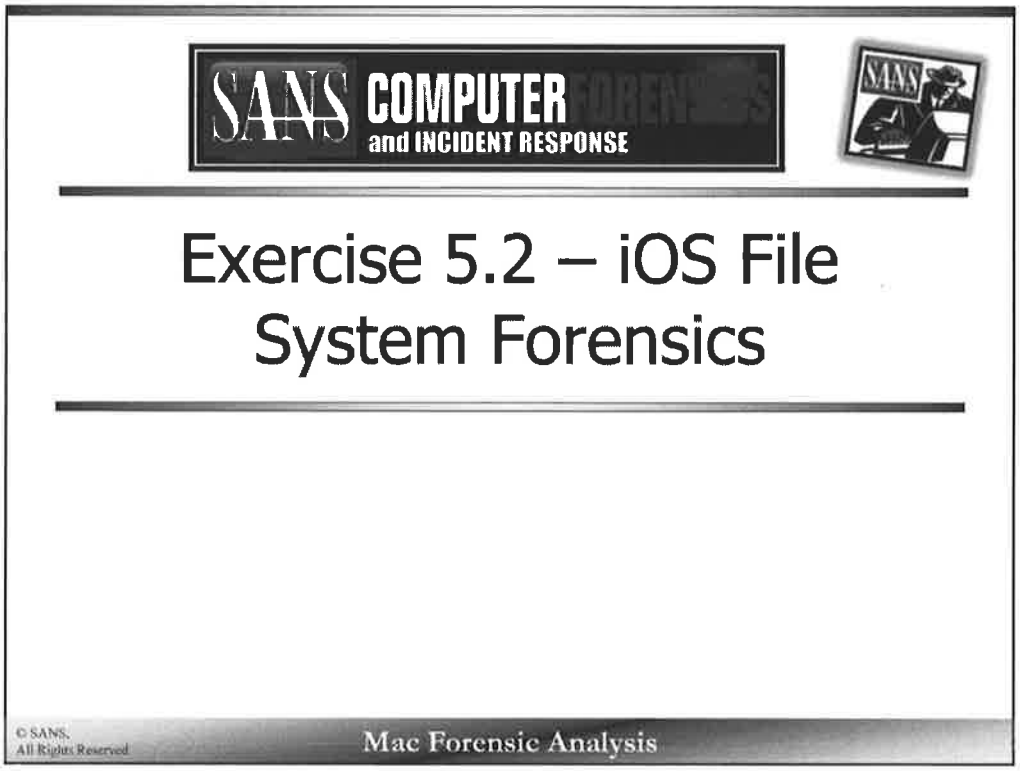




Users can configure "Home" addresses as well as store information for frequently visited locations. Not all applications store data the same way as noted by the coordinates stored by the Waze application above.


search_conf 2013-08-31 (UTC)
session 2013-08-31 (UTC)
shields_conf 2013-08-31 (UTC)
user
version
Library
Cookies
Cookies.binarycookies
Preferences
com.waze.iphone.plist
SMS
sms.db
Google
38.882221, -77.106910
Get directions
My places
3900 Fairfax Dr
Arlington, VA 22209 20 ft NW
Directions
Search nearby
Save to map
more
heb.Update time: Tue, 20 Aug 2013 14:
eng.Update time: Sat, 14 Sep 2013 08:44:58 GMT
Navigation.Last position: -77106910,38882221
Navigation.Last dest name: Home
Navigation.Is navigating: 0
Navigation.Event: 0

214



This page intentionally left blank.