Modern smartphone forensics

Apple iCloud (backups, FindMyPhone, document storage)
encrypted BlackBerry backups (BB 10 and Olympia Service)
Windows Phone 8 (yet another cloud for backups)
Global smartphone market

- About 1.2 billion smartphones worldwide
- "Smart devices" – carry a lot of sensitive data
- Corporate deployments are increasing
- ... hard need for forensics!

(Source: IDC Worldwide Quarterly Mobile Phone Tracker)
## Smartphone forensics methods

<table>
<thead>
<tr>
<th></th>
<th>Android</th>
<th>iOS</th>
<th>Windows Phone</th>
<th>BlackBerry OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical acquisition</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>?</td>
</tr>
<tr>
<td>Physical acquisition</td>
<td>Yes/No</td>
<td>Yes/No</td>
<td>No</td>
<td>Yes*</td>
</tr>
<tr>
<td>Chip-off</td>
<td>Yes/No</td>
<td>No</td>
<td>?</td>
<td>Yes</td>
</tr>
<tr>
<td>Local backup</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Cloud backup</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Documents in cloud</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Location service</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Backups to cloud: why?
iOS forensics
- logical & physical acquisition

• Logical acquisition
  • “Ask” device to produce backup
  • Device must be unlocked (by passcode or iTunes)
  • Device may produce encrypted backup
  • Limited amount of information

• Physical acquisition
  • Boot-time exploit to run unsigned code or jailbreak
  • Device lock state isn’t relevant, can bruteforce passcode
  • Can get all information from the device
  • ... but not for iPhone 5 and iPad 4 :(
iOS passcode

- Device passcode
  - Protect unauthorized access to the device
  - Bypassing is not enough (used in encryption)
- Disk encryption
- Keychain
  - System-wide storage for sensitive data (keys, passwords etc)
  - Data is encrypted
iCloud services

- Introduced in Oct 2011
- Introduced with iOS 5
- 5 GB free storage
- Up to 50 GB paid storage
- Over 320 million users in July 2013
- Backups, documents, notes, calendar, Find My Phone
iCloud backup - what & when

- Contacts and Contact Favorites
- Messages (including iMessages)
- Call history
- Application data
- Device settings
- Camera roll (photos and videos)
- Purchases (music, movies, TV, apps, books)
- Mail accounts
- Network settings (saved Wi-Fi hotspots, VPN settings etc)
- Paired Bluetooth devices
- Offline web application cache/database
- Safari bookmarks, cookies, history, offline data
- ... and much more

- Backup runs daily when device is:
  - Connected to the Internet over Wi-Fi
  - Connected to a power source
  - Locked
- Can force backup
  - [Settings] | [iCloud] | [Storage & Backup] | [Back Up Now]
iCloud backups reverse-engineering

- jailbreak iPhone
- Install Open SSH, get keychain (keychain-2.db)
- [Settings] | [iCloud] | [Delete Account] | [Delete from My iPhone]
- [Settings] | [General] | [Reset] | [Reset All Settings]
- reboot
- set up Wi-Fi connection (proxy)
- replace keychain with our own trusted root certificate (need key 0x835 & keychain)
- ... read all the traffic :)
iCloud backup protocol flow

• Dynamic: endpoints depend on Apple ID

• Built on Google Protocol Buffers (mostly)

• Files are split into chunks

• Apple provides file-to-chunks mapping, chunk encryption keys, and full request info to 3rd-party storage provider (Amazon/Microsoft)

• Encryption key depends on chunk data
Files in iCloud
iCloud backup: authentication

query:
https://setup.icloud.com/setup/authenticate/$APPLE_ID$,

Authorization: Basic <authentication data>
authentication data = mime64 (AppleID:password)

returns: mmeAuthToken, dsPrsID

example:
GET /setup/authenticate/$APPLE_ID$ HTTP/1.1
Host: setup.icloud.com
Accept: */*
User-Agent: iCloud.exe (unknown version) CFNetwork/520.2.6
X-Mme-Client-Info: <PC> <Windows; 6.1.7601/SP1.0; W> <com.apple.AOSKit/88>
Accept-Language: en-US
Authorization: Basic cXR0LnRld3RAaWNtb3VkLmNvbTqRd2VydHkxMjM0NQ==
iCloud backup:
get auth. token, backup IDs, keys

query:
https://setup.icloud.com/setup/get_account_settings
Authorization:Basic <authentication data>
authentication data = mime64 (dsPrsID:mmeAuthToken)
returns: mmeAuthToken (new/other one!!)

query:
https://p11-mobilebackup.icloud.com/mbs/(dsPrsID)
Authorization: <authentication data>
authentication data = mime64 (dsPrsID:mmeAuthToken)
returns: list of backup IDs (backupudid)

query:
https://p11-mobilebackup.icloud.com/mbs/2005111682/(backupudid)/getKeys
Enumerate snapshots

HTTPS GET
https://p11-mobilebackup.icloud.com/mbs/(dsPrsID) / (backupudid)/(snapshotid)/listFiles?
offset=(offset)&limit=(limit)

Get file authentication tokens

HTTPS POST
https://p11-mobilebackup.icloud.com/mbs/(dsPrsID)/(backupudid)/(snapshotid)/getFiles
iCloud backup: download files (2)

Download chunks

Windows Azure:
http://msbnx000004.blob.core.windows.net:80/cnt/g6YMJKQBxQruxQAr30C?sp=r&sr=b&byte-range=154-31457433&se=2013-06-07T10:14Z&st=2013-06-07T09:19Z&sig=0EdHy75gGHCee%2BjKePZBqz8xbWxpTxaYyASwFXVx2%2Fg%3D

'se' contains iCloud authorization time (expires in one hour)

Amazon AWS:
http://us-std-00001.s3-external-1.amazonaws.com/l9rh20QBPX4jizMAr3vY?x-client-request-
iCloud encryption

- Data stored at 3rd-party storage providers is encrypted
- Apple has encryption keys to that data
- Few files are further encrypted using keys from OTA backup keybag
- Keychain items are encrypted using keys from OTA backup keybag
- Need key 0x835 (securityd) to decrypt most keys from OTA backup keybag

iCloud backups - summary

- There is no user-configurable encryption for iCloud backups
- iCloud backups are stored in Microsoft and Amazon clouds in encrypted form
- Apple holds encryption keys and thus have access to data in iCloud backups
- **If Apple stores 0x835 keys then it can also have access to Keychain data (i.e. passwords)**
- Apple may have legal obligations to do this (e.g. LE)
- No notification after backup downloading (as with device restore)
Find My Phone
FindMyPhone protocol

How: just sniffing HTTP traffic (www.icloud.com, Find My Phone)

Authorization:

validate:
https://setup.icloud.com/setup/ws/1/validate

ClientBuildNumber=1M.63768 (constant)
ClientId (random GUID)
<- instance

login:
https://setup.icloud.com/setup/ws/1/login

AppleID
extended_login
id=sha1(apple_id+instance)
password
<- dsid

Get devices with location:

initClient:
https://p11-fmipweb.icloud.com/fmipservice/client/web/initClient

refreshClient:
https://p11-fmipweb.icloud.com/fmipservice/client/web/
refreshClient

id
dsid
<- content (location)
FindMyPhone - demo output

deviceModel = SixthGen-white
modelDisplayName = iPhone
id = QVBTOmTwmI1ZiMDY5ODg5NDa2MGtwMDhjQWY4MWRkOWU2YWItM2UwN2JiYWY-
deviceDisplayName = iPhone 5
name = Vladimir's iPhone 5
batteryLevel = 0.700216
locationEnabled = 1
longitude = 37.6243
latitude = 55.8114
positionType = wifi
isOld = 0
Device found: 00:00:27.786000 ago

deviceModel = MacBookAir3,2
modelDisplayName = MacBook Air
id = QVBTOjQ3MzFENjQ5LTDGRiMkNTkzRG1CMzIjLUZCMkRFNjA2MjRGRRA~~
deviceDisplayName = MacBook Air 13"
name = Vladimir Katalov's MacBook Air
batteryLevel = 0
locationEnabled = 1
longitude = 0
latitude = 0
positionType =
isOld = 0
Device not found

deviceModel = ThirdGen-4G
modelDisplayName = iPad
id = QVBTOiAzYjuZyJiMiZzZVFHzVmE3NWU2MTk5YzQzYzQwNDJjYTJjNjRkZig~
deviceDisplayName = iPad
name = Vladimir Katalov's iPad
batteryLevel = 0.74746
locationEnabled = 1
longitude = 37.6245
latitude = 55.8113
positionType = Wifi
isOld = 0
Device found: 00:00:36.485000 ago
iCloud documents
Get files from iCloud

To get list of files

- Authentication request (with given AppleID & password). Client gets mmeAuthToken in return; which, in order, is used to create authentication token (together with dsid). dsid (Destination Signaling IDentifier) is a unique ID assigned to the user when registering at iCloud.com.

- Request to get AccountSettings. Client gets an URL (ubiquityUrl) with an address to get UUID (unique user identifier), file list, info on file tokens and for authorization.

- Request to get file list (POST). Output (for every file):
  - file name
  - file id
  - parent folder id
  - last change time
  - checksum
  - access rights

To download given file

- Request to get file token (using file id, checksum and aliasMap).

- Authorization request. Returns information on file chunks and containers. Output: container list (with URLs) and chunk information.
iCloud backup: packages

- KeyNote: PDF, Microsoft PowerPoint, KeyNote ’09
- Pages: PDF, Microsoft Word, Pages ’09
- Numbers: PDF, Microsoft Excel, Numbers ’09
- Some other programs (1Password etc)

Storage: plist + content (text, media files)

Requests:
<table>
<thead>
<tr>
<th>File name</th>
<th>File path</th>
<th>File id</th>
<th>File size</th>
</tr>
</thead>
<tbody>
<tr>
<td>buildVersionHistory.plist</td>
<td>/com-apple-Pages/Documents/EIFT FAQ.pages.tef/buildVersionHistory.plist</td>
<td>422124650662430</td>
<td>221 bytes</td>
</tr>
<tr>
<td>index.db</td>
<td>/com-apple-Pages/Documents/EIFT FAQ.pages.tef/index.db</td>
<td>422124650662429</td>
<td>376382 bytes</td>
</tr>
<tr>
<td>index_viewstate</td>
<td>/com-apple-Pages/Documents/EIFT FAQ.pages.tef/index_viewstate</td>
<td>422124650662435</td>
<td>713 bytes</td>
</tr>
<tr>
<td>metadata.plist</td>
<td>/com-apple-Pages/Documents/EIFT FAQ.pages.tef/metadata.plist</td>
<td>422124650662431</td>
<td>416 bytes</td>
</tr>
<tr>
<td>preview-micro.jpg</td>
<td>/com-apple-Pages/Documents/EIFT FAQ.pages.tef/preview-micro.jpg</td>
<td>422124650662442</td>
<td>1489 bytes</td>
</tr>
<tr>
<td>preview-web.jpg</td>
<td>/com-apple-Pages/Documents/EIFT FAQ.pages.tef/preview-web.jpg</td>
<td>422124650662443</td>
<td>11782 bytes</td>
</tr>
<tr>
<td>preview.jpg</td>
<td>/com-apple-Pages/Documents/EIFT FAQ.pages.tef/Previews/preview.jpg</td>
<td>422124650662444</td>
<td>45229 bytes</td>
</tr>
<tr>
<td>EIFT FAQ.jpg</td>
<td>/com-apple-Pages/iWorkPreviews/EIFT FAQ.jpg</td>
<td>422124650662514</td>
<td>45229 bytes</td>
</tr>
</tbody>
</table>
Apple 2FA (two-step verification)

Enter Verification Code.
We have sent a temporary verification code to John's iPod touch. Enter the code to continue. Didn't receive a code?

Verification Code:

Send
Send a new code
Cancel
Continue

Verification Code
Your Apple ID verification code is: 6985
Apple 2FA
(cont-d)

Requires to verify your identity using one of your devices before you can:

• Sign in to My Apple ID to manage your account.
• Make an iTunes, App Store, or iBookstore purchase from a new device.
• Get Apple ID-related support from Apple.

Does NOT protect:

• iCloud backups
• Find My Phone data
• Documents stored in the cloud
Apple iOS 7
iCloud keychain
iCloud keychain
Apple iOS 7
iCloud keychain - cont-d

Enter your iCloud Security Code.

Verification Code sent to 7 (985) 998-6820
Enter Verification Code

To verify your identity, a message was sent to the phone number Apple has on file.

Allow "Mac Mini" to use iCloud Keychain
Enter the Apple ID password for "vkatalov@mail.ru" to allow Mac Mini to use your passwords.
Apple iCloud: Conclusion

- Balance between security, privacy and convenience
- iCloud security risks
- Use additional encryption
- Better 2FA implementation
- Need further work
  - My Photo Stream
  - Photo Sharing
  - 3rd party apps data
  - New security classes
  - iCloud keychain
  - Back To My Mac
  - FindMyPhone on iOS 7
  - Touch ID (iPhone 5S)
Windows Phone backups

What is saved:

• Internet Explorer Favourites
• List of installed apps
• Theme and accent configuration
• Call history
• App settings (where applicable - email and accounts, lock screen etc)
• Test messages (SMS conversations)
• Photos (good quality - uses data allowance)

Can get with LiveSDK:

• Basic user information
• Contacts
• Calendars
• Files, photos, videos, documents

Download full backup?
Windows Phone: Live SDK

Identity API

• Get basic information on user

Hotmail API:

• Manage contacts
• Manage calendars & events

SkyDrive API

• Files & documents
• Photos
• Videos
Windows Live SDK (cont’d)

• Authentication
  • Needs client_id of registered application
  • Several requests to https://login.live.com to get redirects and some parameters
  • Get antiForgeryVerificationToken
  • Get access_token

• Get basic info
  GET https://apis.live.net/v5.0/me?access_token=…

• Get contacts
  GET https://apis.live.net/v5.0/me/contacts?access_token==…

• Get access to SkyDrive
  GET https://apis.live.net/v5.0/me/skydrive/my_documents?
  access_token==…
WP8: get SMS

- Server: https://???-m.hotmail.com
  (to get correct name: send request to blum-hotmail.com)
- Protocol: ASHTTP
- Data format: wbxml
- Can be compressed ("Accept-Encoding: gzip, deflate")

Requests/responses:
- Get (login, password in base64
  ★ success
- FolderHierarchy
  ★ success
- SyncKeys
  ★ success
- CategoriesItems (CollectionId)
  ★ return: SyncKey
- GetItemEstimate (SyncKey, CollectionId)
  ★ number of SMS
- GetSMS (SyncKey, CollectionId)
  ★ new SyncKey and SMS
- GetClosedSMS
WP8: get mail

GET http://mail.live.com/ HTTP/1.1
Host: mail.live.com
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116 Safari/537.36
Accept-Encoding: gzip, deflate, sdch

< redirect to authentication (https://login.live.com/login.srf?...)

> POST https://login.live.com/ppsecure/post.srf?... (login, password)

< redirect to mail.live.com?id=XXX

> GET https://mail.live.com/?id=XXX

< redirect to mailbox:

HTTP/1.1 302 Found
Location: https://col131.mail.live.com/default.aspx?id=XXX&rru=inbox

Go to mailbox:

GET https://col131.mail.live.com/default.aspx?id=XXX&rru=inbox HTTP/1.1
Host: col131.mail.live.com
Connection: keep-alive
WP8: find my phone

- Map phone’s location
  - GET https://www.windowsphone.com/ru-ru/my/find HTTP/1.1
  - or
- Make the phone ring
  - https://www.windowsphone.com/ru-ru/my/phones/c34a5c89b6aabc87cdc457b49e5f3abbbf81c72e0b19d48bd8d3918e36785f646/ring
- Lock the phone and show a message
- Erase the phone

Authentication is required, of course!

You can set up Find My Phone to save your location every few hours or to use push notifications instead of text messages to send commands (and apps)
BlackBerry backups

Old format:

- IPD files (all databases in a single container)
- BBB files (in fact, ZIP archives with several IPDs, one database per IPD)

New format:

- Unencrypted BBB-QNX (three .tar files inside); for PlayBook with firmware <2.0
- Encrypted BBB-QNX (all .tar files are encrypted); for BB OS 10 (backup created with BlackBerry Link)

For old formats - simple password protection:

- Encryption: AES-256
- Password verification:
  - BlackBerry Desktop Software 5: pbkdf2 (1) - yes, just one iteration
  - BlackBerry Desktop Software 6: pbkdf2 (20,000)
BB10 backups

- mounting QNX6 partitions
- backup encryption: AES-256
- authentication/verification: HMAC-SHA1
- backup.cgi:backuparch
- backup.cgi:scramble

- bbid (BlackBerry ID)
  (libbbid.so:bbid_profile_get_user_properties(urn:bbid:username))

- qbek
  (libbbid.so:bbid_profile_get_user_properties(urn:bbid:backupandrestore key))

- cache storage: /accounts/<id>/sys/bbid/keyCache

- if not found: request to BB Olympia Service
  (blackberryid.blackberry.com)
BlackBerry Token Service

- request: bbid, password, pin, salt (client's entropy)
- response:

```
Hct=1379081439336&st=1379168703336&se=PF3V5ikbH8fx2wSb2mbHITGy0q1xIcGZ66Oma3o66k&et=1381673439336
&fn=John&ln=Doe&nn=johndoe-59094&un=john.doe%40gmail.com&ec=AcDGzWbVM12nd0BiqgfJYw%3D&em=john.doe
%40gmail.com&at=AQ:AQ:zTh0_L5BwTuZf0w0L2CYVGmMryzSbs7OsPBq7ZiYY:ibKt2ZKG0sAjiODk6lITmQA:asSsJ
MYRzS8Tf2IMQY44_HiCDaWzCBRwQj68XDDHOs6Qhp7gCXuKqSk6_v4KTQ8pWMtpVriBNBWO4tlq879MY_Oro2upCz
w32EmCgAKapUPG7leAIKeo3kr13v-Td2lpWU0b3kQJVJstMz9GBjJ29RFkcxw-039ksxUJYnDxkCrgbrAwVFpw5Pg5XmAZxtA
```
- se - server entropy
- at - authentication token
- ec - user ID for BB cloud services (saved to /dev/rpmb/BBID_BDEK)
- at (creation time), st (server time), et (expirity time)
- further requests: RST (Request Secure Token) with token type and service name
- to get qbek:
  - get authentication token
  - get BBIDAuthN_1 token for urn:bbid:v1:olympia
  - send request for authzo:qbek token
  - register device on BB server (using authzo:qbek token)
  - get request on backupAndRestoreKey info (two IDs)
  - get janusUrl by request to kronos.bbprotect.blackberry.com
  - get qbek from %janusUrl%/FlashGetFile
BB10 device

bbidd

userCache

keyCache

QbekRequest

Qbek

Authentication(deviceSerialNumber, usr, pwd)

tokenSecretKey

QbekRequest(AES(request, tokenSecretKey), bbidd)

qbek

WritePreamble(tagType, iterCount, salt)

WriteBlocks(SHA(block, KDF(qbek, salt, iterCount)))

Resultant backup file

Olympia
Thank you!

Modern smartphone forensics

Vladimir Katalov, ElcomSoft Co. Ltd.
(twitter: @vkatalov)

http://www.elcomsoft.com
http://blog.crackpassword.com