Do-It-Yourself Mobile Forensics

Lewis Sykalski

lsykalski@smu.edu
Contents

1. Digital Forensics ........................................................................................................... 4
   1.1 Mobile Forensics ................................................................................................. 4
   1.2 Forensic Recovery Principles ............................................................................. 4
2 Mobile Forensics Recovery Techniques ..................................................................... 4
   2.1 Physical Techniques ......................................................................................... 5
      2.1.1 JTAG recovery .......................................................................................... 5
      2.1.2 Chip-Off Technique ............................................................................... 5
      2.1.3 Software Physical Image Capture ........................................................... 5
   2.2 Logical Techniques ............................................................................................ 5
      2.2.1 Backup Analysis ..................................................................................... 6
      2.2.2 ITunes Backup Analysis .......................................................................... 6
      2.2.3 Android Backup Analysis ....................................................................... 7
      2.2.4 Live Extraction ....................................................................................... 7
3 Commercial Off-The-Shelf (COTS) Tools ................................................................. 7
   3.1 ViaForensics ViaExtract ................................................................................ 7
   3.2 Conclusion on Commercial Tools ..................................................................... 9
4 Free/Open Source Tools ............................................................................................. 9
   4.1 The Sleuth Kit (TSK) ....................................................................................... 10
      4.1.1 Autopsy .................................................................................................. 10
      4.1.2 Tools in TSK ......................................................................................... 10
      4.1.3 A Sample Forensic Recovery Using Autopsy ....................................... 10
      4.1.4 Conclusions on Autopsy/TSK ................................................................ 16
   4.2 IPhone Backup Analyzer .................................................................................. 16
      4.2.1 IPhone Backup Analyzer Case Study .................................................... 17
      4.2.2 IPhone Backup Analyzer Conclusions .................................................. 22
   4.3 iOS Forensic Tools ............................................................................................ 22
   4.4 System Tools/Commands .................................................................................. 22
5 Summary ..................................................................................................................... 23
6 References .................................................................................................................. 24
Table of Images

Image 1: Sample JTAG layout ................................................................. 5
Image 2: VirtualBox Virtual Machine ...................................................... 8
Image 3: NABI2 Tablet ...................................................................... 8
Image 4: viaExtract - Creating a new Case ......................................... 8
Image 5: Autopsy: Mounting Image ...................................................... 11
Image 6: Autopsy: Verifying the MD5 Sum .......................................... 12
Image 7: Autopsy File Analysis Pane .................................................. 13
Image 8: Jimmy Jungle.doc ................................................................. 13
Image 9: Autopsy Data Unit Pane ......................................................... 14
Image 10: Finding goodtimes password ................................................ 15
Image 11: Coverpage.jpg .................................................................... 15
Image 12: Keyword Search Using Autopsy .......................................... 16
Image 13: iPBA main-screen ............................................................... 18
Image 14: GPS Coordinates of Monica on December 18th, 2010 ......... 19
Image 15: Google Maps Location of Monica on 12/18/2010 ................ 19
Image 16: Safari Bookmarks ............................................................... 20
Image 17: Suspicious SMS message .................................................... 21
Image 18: Suspicious Contact ............................................................. 21

Table of Tables

Table 1: iPhone Backup File Information ................................................ 7
Table 2: COTS Mobile Forensic Tools .................................................... 7
Table 3: viaExtract Browser History ..................................................... 9
Table 4: Free/OSS Mobile Forensic Tools ............................................. 10
Table 5: HoneyNet Scan 24 Questions .................................................. 11
Table 6: Condensed Scheduled Visits.xls .............................................. 15
1. Digital Forensics

Digital Forensics is a growing field that most law-enforcement, corporations, and attorneys take part in each and every day. Largely speaking, it surrounds the retrieval and subsequent investigation of evidence or other material found in digital devices, and has become a key aspect to all types of investigations, whether it be criminal, internal, or private. Digital Forensics is further divided into 4-subfields, namely computer forensics, database forensics, mobile forensics, and network forensics.

The general population is still largely unaware of the existence of the field, however many people, including criminals, might benefit from even a cursory understanding as it might help them to recover digital information or prevent such recovery. Whether it’s spying on a suspected unfaithful spouse or investigating an employee as a small-business owner, there are many do-it-yourself measures to recover information needed to validate or terminate relationships in your personal or professional lives. This paper seeks to define mobile forensics and the various strategies for accomplishing forensic acquisitions. It also seeks to survey the various tools available for a do-it-yourselfer.

1.1 Mobile Forensics

Mobile Forensics is a branch of digital forensics. It governs the recovery of material from a mobile device, typically a phone, tablet, PDA, or GPS device, however it could also include other embedded devices with both communication ability and internal memory. The type of data recovered with mobile forensics is usually wide-ranging including e-mail, contacts, photos, notes, text messages, recordings, calendars, video, web-browsing history, and social network communications.

The advances in smart-phone technology are having impact on the field. Mobile forensics is now becoming key to an increasing percentage of investigations and bringing to the forefront new sets of software tools & recovery techniques. Additionally, over the last 10-12 years, mobile forensics has fast become a $300 million dollar-a-year industry in the United States as corporations and law enforcement have spent substantial amounts of money looking for mobile-forensic solutions.

1.2 Forensic Recovery Principles

The Association of Chief Police Officers (ACPO) has established 4 principles that should be adhered to when completing a forensic investigation on computer-based electronic evidence [1]:

1. No action taken by law enforcement agencies or their agents should change data held on a computer or storage media, which may subsequently be relied upon in court.
2. In circumstances where a person finds it necessary to access original data held on a computer or on storage media, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions.
3. An audit trail or other record of all processes applied to computer-based electronic evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result.
4. The person in charge of the investigation (the case officer) has overall responsibility for ensuring that the law and these principles are adhered to.

Although there may be different flavors of the afore-mentioned principles, the spirit of the principles is always the same – namely do everything in your power to not risk corruption of the evidence!

2 Mobile Forensics Recovery Techniques

Mobile Forensics recovery strategies can either be hardware (physical) or software (logical) based. Hardware techniques include desoldering chips and using JTAG interfaces to do full-scale memory reads. Software techniques include using Commercial Off-The-Shelf (COTS) Software, Open-Source Software, System-Command Line Tools, or custom commercial/government investigation techniques. Investigators are finding there is no 1-size fits all solution and that no-1 tool or technique provides sufficient utility to acquire full-coverage.
2.1 Physical Techniques

Physical techniques provide for direct access to the physical storage medium. Since it does not go through the O/S to access the data, unallocated data (i.e. deleted files) in addition to allocated data can be recovered.

2.1.1 JTAG recovery

The JTAG interface standard was created in the 1980s in order to concoct an associated standard for validating the wiring and interconnects on printed circuit boards (PCB). By the year 1990, the standard was finished (IEEE 1149.1-1990). This was followed, in turn, by a later update in 2001 -- IEEE 1149.1-2001. The standard is pretty widely accepted today. Most PCBs have JTAG test access ports (TAPs) that can be used to extract information from memory & the CPU.

In order to perform a JTAG recovery, one must connect the appropriate JTAG pins (Image 1 below) to the memory flasher. Once the board is powered on, the flasher software can then a full memory dump of the NAND flash. This takes a significant amount of time. The connections can then be broken and the phone can be reassembled.

Although this captures a full physical image, it is not normally used as logical means can perform sufficient coverage. Also, any errors in soldering or voltages can destroy the very small PCB connections & the device itself.

![Image 1: Sample JTAG layout](image1.png)

2.1.2 Chip-Off Technique

A Chip-Off Technique is just as the name implies, full removal of the chip (in this case the NAND flash chip). The Chip-Off technique provides a method for recovery of already damaged devices (e.g. water damage). It also provides for circumvention of any pass-coded devices where the pass-code or pin is unknown.

The removal of the chip is generally irreversible, as it is quite difficult to resolder/reattach the flash chip to the PCB successfully. In order to perform this technique, the flash chip is desoldered using hot-air/vacuum to remove the chip. The chip is then inserted into a flash reader; wherein-the-device must be programmed to read the particular variant of chip in question.

2.1.3 Software Physical Image Capture

Many devices support live capture via USB in order to capture a physical image of the device. This technique allows a software application to run with root access on the target device in order to provide a full physical image of the data partitions. This has the advantage over the hardware options as it is easier to perform. Because it is a physical capture, it provides direct access to the file systems to allow a complete copy of all files. It also results in very little risk of damaging the device or resultant data loss.

2.2 Logical Techniques

Logical techniques extract allocated data through software. This is generally achieved by accessing the file system (O/S) to extract primarily the allocated data.
2.2.1 Backup Analysis

Forensic analysis on an active device will undoubtedly result in rebooting of the target system, and as such may alter the contents stored on the device. In criminal investigations, forensic examiners only operate on backups. For the iPhone, this could be a logical backup acquired through iTunes. For the android this could be a backup from Google drive or an image of the disk.

Also, backup analysis can be performed without even having access to the device in-question. This is because often people leave backups resident on their personal computers and don’t delete them. They are often also very lax with ensuring this information is adequately safe-guarded, foregoing encryption/password strategies in creating the backup.

2.2.2 iTunes Backup Analysis

AFC (Apple file connection) protocol is used by iTunes to create the backup. The backup process is lossless and has low-corruption risk as it does not alter anything on the iPhone except escrow key records.

The user has the option in performing an iTunes backup to encrypt the backup with a password of his choosing. After choosing such password, his/her files are backed-up and subsequently encrypted with the key using AES-256 in CBC mode (Cipher-Block-Chaining). The Encryption keys themselves are stored in the Backup Keybag and protected by iTunes password that the user supplied earlier. As such, decryption of files in the newly created encrypted backup is not possible without the previously supplied iTunes password. Because iTunes does not require have any password strength requirements or impose any limitations on the password itself, it is somewhat easy to perform a brute force attack on the password. As such, encrypted backups may be more difficult to recover or even impossible when a complex password was used.

The information contained in an iTunes back is pretty much everything on the device (i.e. contacts, photos, SMS text messages, calendar, music, call logs, bookmarks, cookies, app data, etc.) A list of the file-system location of some of the more important pieces of information is shown below in Table 1. In addition to these items, iTunes also backs up identifying markers, namely the SIM hardware number, phone number, UDID, and serial number. It does not, however, save any unallocated deleted files.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddressBook.sqlitedb</td>
<td>Contact information and personal data like name, email address, birthday, organization, etc...</td>
</tr>
<tr>
<td>AddressBookImages.sqlitedb</td>
<td>Images associated with saved contacts</td>
</tr>
<tr>
<td>Calendar.sqlitedb</td>
<td>Calendar details and events information</td>
</tr>
<tr>
<td>Call_history.db</td>
<td>Incoming and outgoing call logs including phone numbers and time stamps</td>
</tr>
<tr>
<td>Sms.db</td>
<td>Text and multimedia messages along with their timestamps</td>
</tr>
<tr>
<td>Voicemail.db</td>
<td>Voicemail messages</td>
</tr>
<tr>
<td>Safari/Bookmarks.db</td>
<td>Saved URL addresses</td>
</tr>
<tr>
<td>Safari/History.plist</td>
<td>User’s internet browsing history</td>
</tr>
<tr>
<td>Notes.sqlite</td>
<td>Apple Notes application data</td>
</tr>
<tr>
<td>Maps/History.plist</td>
<td>It keeps track of location searches</td>
</tr>
<tr>
<td>Maps/Bookmarks.plist</td>
<td>Saved location searches</td>
</tr>
<tr>
<td>consolidated.db</td>
<td>Stores GPS tracking data</td>
</tr>
<tr>
<td>En_GB-dynamic-text.dat</td>
<td>Keyboard cache</td>
</tr>
<tr>
<td>com.apple.accountsettings.plist</td>
<td>Maintains data about all email accounts that are configured on the Apple Email application</td>
</tr>
<tr>
<td>com.apple.network.identification.plist</td>
<td>Wireless network data including IP address, router IP</td>
</tr>
</tbody>
</table>
2.2.3 Android Backup Analysis

Android, when it first came out, did not have a central backup program like iPhone does. As a result, many backup apps were created. Since this time, backup APIs have been written and are more tightly integrated with apps; however no central backup app exists. As a result, forensic analysts need to determine what, if any, backup app has been installed or is being used & where this data is being stored (cloud, another host machine, sd-card, etc)?

2.2.4 Live Extraction

Many devices & applications support live capture via USB or other protocol. For example, the Android Debug Bridge (ADB) is often used as a means for forensic analysis capture. ADB is found as part of the Android SDK Platform-Tools package, and consists of both client & server-side programs made to communicate with one-another. [2]: ADB uses a command-line interface which forensic programs are using to capture the necessary information from the device.

3. Commercial Off-The-Shelf (COTS) Tools

There are a wide-range of commercial tools available these days, each having unique applications/strategies in order to recover data as well as target systems. Major players in the industry include Cellebrite, viaForensics, MicroSystemation, Paraben, Radio Tactics, Oxygen Software Company, and Compelson among others with no 1 company dominating the market share. The decision to use these tools should be balanced with the cost & relative utility that each provide while keeping in mind that Free/Open-Source Options are available that may/may not provide similar capability. A short description of some of these tools is provided below in Table 2

<table>
<thead>
<tr>
<th>Company</th>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cellebrite [6]</td>
<td>Universal Forensics Extraction Device</td>
<td>Standalone-portable analyzer that connects to mobile device performing a wide-variety of functions: password/data extraction</td>
</tr>
<tr>
<td>Compelson [4]</td>
<td>Mobiledit! Forensic</td>
<td>Software tool that connects to portable device via Bluetooth or other cable connection. Generates pre-formatted reports on extracted password &amp; other data that is resident on the device</td>
</tr>
<tr>
<td>MicroSystemation [5]</td>
<td>XRY/XACT</td>
<td>Software application which allows user to perform a secure forensic extraction of data (either logical or physical) from a mobile device. Also offers military-specified ruggedized laptop.</td>
</tr>
<tr>
<td>Oxygen Software Company [7]</td>
<td>Forensic Suite</td>
<td>Software application which provides for basic extraction of logical data from device specifically designed for smart-phones</td>
</tr>
<tr>
<td>Paraben [8]</td>
<td>Device Seizure</td>
<td>Software package that performs both logical and physical extractions. Also offers military-specified ruggedized laptop.</td>
</tr>
<tr>
<td>Radio Tactics [9]</td>
<td>Aceso</td>
<td>Kiosk &amp; Field-Device that performs comprehensive data extraction from mobile phones, gps devices, and sim-media cards</td>
</tr>
<tr>
<td>viaForensics [10]</td>
<td>viaExtract</td>
<td>Data-Extraction Software specifically designed for android-powered portable devices</td>
</tr>
</tbody>
</table>

Table 2: COTS Mobile Forensic Tools

3.1 ViaForensics ViaExtract
I decided to evaluate a commercial tool as part of this research paper. ViaForensics seemed the logical choice, so I got device-coverage. (I later analyze an iPhone backup). 

ViaForensics performs a live logical capture through USB of host machine to device. In order to run viaExtract, however, it requires a Virtual Machine be installed. The two options provided were Virtual Box & VMWare. After going through the process of installing Virtual Box, setting appropriate group permissions, and installing the custom application software on the VM, I decided to connect up our kids NABI 2 tablet to the host machine and do a live capture.

![Image 2: VirtualBox Virtual Machine](image2.png) ![Image 3: NABI2 Tablet](image3.png)

After defining the case information, loading the data from the device, and defining the contents of the report, I hit export and was presented with a full colorful data-dump of the important contents of the device – namely, what websites they hit, bookmarks, searches, etc.

![Image 4: viaExtract - Creating a new Case](image4.png)
3.2 Conclusion on Commercial Tools

Commercial tools are of obvious value and should be considered due to robustness & ease of use. The reports are generally of a higher caliber than any free/open source product. Also, there is generally a higher degree of trust for commercial products in the law-enforcement world, so this should be considered if one intends to build an active case. Additionally, there is also an inherent cost involved (e.g. $1500 for a viaExtract license), which should be considered before purchasing anything. Also, certain tools are available for free or at a discounted price to law enforcement, so if applicable this should be researched.

4 Free/Open Source Tools

While commercial tools exist, there are a growing number of Free/OSS tools. A Mac or Linux workstation is used for many of these programs. These may be a better option for the do-it-yourselfer or the more price-conscious.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Target System</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFLogical OSE [11]</td>
<td>Android</td>
<td>Software tool that analyzes Android via uss connection. Must use vmware, however significant targeted information can be pulled from device with simple button clicks (currently government only)</td>
</tr>
<tr>
<td>iOS Forensic Tools [12]</td>
<td>IPhone</td>
<td>Zdiarksi’s iPhone Forensic Tools, physical/logical acquisition, pin bypass, crack encrypted data</td>
</tr>
<tr>
<td>iPhone Backup Analyzer [13]</td>
<td>IPhone</td>
<td>Software tool that analyzes an iOS backup (iPhone, iPad, etc). Pretty much all information can be recovered from device in a friendly user-interface format.</td>
</tr>
<tr>
<td>Lantern Lite [14]</td>
<td>IPhone</td>
<td>Performs physical data forensic extractions (currently government only)</td>
</tr>
<tr>
<td>Mobile Internal Acquisition Tool [15]</td>
<td>Any</td>
<td>Performs physical extraction without using external hardware</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>The Sleuth Kit (TSK)/Autopsy [16]</td>
<td>Any</td>
<td>Open Source tool written by Brian Carrier used for forensic analysis. HTML-Server provides for a Browser Front-End to the tools if desired</td>
</tr>
<tr>
<td>System tools</td>
<td>Either</td>
<td>There are a number of Linux system provided tools (file, md5sum, grep, mount, etc) which help in forensic analysis</td>
</tr>
</tbody>
</table>

Table 4: Free/OSS Mobile Forensic Tools

4.1 The Sleuth Kit (TSK)

The Sleuth Kit (TSK) is a suite of Unix/Windows-based tools, scripts & utilities as well as library APIs that provide for the forensic analysis of computer systems including mobile devices. It is authored and maintained by Brian Carrier. TSK can be used to extract relevant data for investigation purposes from images taken from various types of computers & mobile devices. While all these tools can be used standalone, TSK is typically used in conjunction with a front-end application, Autopsy. In this configuration, it provides for a friendlier user interface.

4.1.1 Autopsy

Autopsy is a web-server that runs local on the analyzing machine. You connect to this web-server via the web browser of your choice by simply inputting the defined URL (i.e. http://localhost:9999/autopsy) in the search bar. After that, Autopsy provides user-interface to open a new case file & mount your image from the suspect machine in order to examine it in-depth.

4.1.2 Tools in TSK

In addition to the front-end provided by Autopsy, additional & redundant command-line tools/capability are provided in the TSK. Some of these tools include:

- **blkls**: displays user-chosen data blocks within a file system
- **disk_stat**: uncovers Host Protected Areas, hidden areas not normally visible to the O/S.
- **ffind**: Searches for all files that point to an input metadata file
- **fls**: shows both allocated & unallocated file names
- **fsstat**: displays file system information about an image
- **ils**: displays all metadata entries
- **mactime**: creates file timeline (creation/access/modification) based upon their associated MAC times

4.1.3 A Sample Forensic Recovery Using Autopsy

In order to evaluate the power of TSK & Autopsy, I decided to check out some sample case-studies over at the honeynet challenge website [17]. I had casually heard about these challenges, but had never participated. I chose scan 24 (http://old.honeynet.org/scans/scan24/) due to my proficiency level. This case study involved analyzing a marijuana-dealer's floppy disk (which could have just as easily have been a sim-card or mobile device) in order to locate the following information below (Table 5):

| Q1 | Who is Joe Jacob’s supplier of marijuana and what is the address listed for the supplier? |
| Q2 | What crucial data is available within the coverpage.jpg file and why is this data crucial? |
| Q3 | What (if any) other high schools besides Smith Hill does Joe Jacobs frequent? |
Q4 For each file, what processes were taken by the suspect to mask them from others?

Q5 What processes did you (the investigator) use to successfully examine the entire contents of each file?

Table 5: HoneyNet Scan 24 Questions

After ensuring that I didn’t tamper with the contents (read-only copy), I was able to mount the image, using autopsy (Image 5). It’s important to note that these same routines can be done command-line; however, using a html front-end allows even the novice to partake in digital forensics.

![Image 5: Autopsy: Mounting Image](image5)

The second step, in any digital forensic investigation is to ensure that what you are working with hasn’t been tampered with; we do this by verifying the md5 hash. I did this by using the built-in Image Integrity pane in Autopsy (however I could have just as easily used md5sum command-line tool in Linux). (Image 6).
After performing a hash-sum to verify the integrity, I went to file-analysis pane (Image 7) to see what may have been present on the floppy at the time it was seized. The pane showed the following files:

- a document file – Jimmy Jungle.doc
- an image file -- coverpage.jpgc
- an executable – Scheduled Visits.exe

The first thing that I observed was that Jimmy Jungle.doc was deleted (perhaps there is something on there that was not intended to be seen?). After selecting Jimmy Jungle.doc there was an option to export, which I did. After opening the file (Image 8), I the documented yielded the answer to Q1.
The other thing that stood out was that the executable was being recognized as a zip file. Thinking I would have similar luck with the renamed .exe file, I exported it; however after trying to open I received an error indicating it may be part of a multi-part archive. Well perusing the Data Unit pane (Image 9), and looking at the raw string content, I observed that the zip extends beyond the initial 2 sectors that were initially assigned to it. It appeared to extend from Sector 104 to Sector 108. As such, I exported the associated sectors to file, renamed to .zip, and miraculously I could open it. Unfortunately, it was password protected.
Well turning my attention back to the image file, it pointed to Sector 451, so I thought I’d have a look in the same Data Unit Tab. Looking at this sector showed it to be unallocated, yet the file oddly existed and was allocated. The size of the file on the File Analysis tab indicated it was 31 sectors (\( \frac{15585 + 511}{512} \)). This size was equivalent to the sectors resident between the document and the zip file (73-103). Looking at this in the Data Unit Tab revealed it was of type jpeg – BINGO! I exported the image file (Image 11); however, no new information was gathered.

Now pouring back over what I gathered thus far, I noticed the following statement in the .doc file: “To open it, use the same password that you sent me before with that file”. So I decided to scan the image file for any embedded strings by viewing “ASCII Strings” (Image 10) and discovered the string “pw=goodtimes”. This could be the .zip file password?
Using goodtimes as the password to the zip was met with success, where-after I extracted the .xls containing the locations/dates of the schools where the sales were occurring (Table 6 below).

<table>
<thead>
<tr>
<th>Month</th>
<th>DAY</th>
<th>HIGH SCHOOLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>April</td>
<td>Monday (1)</td>
<td>Smith Hill High School (A)</td>
</tr>
<tr>
<td></td>
<td>Tuesday (2)</td>
<td>Key High School (B)</td>
</tr>
<tr>
<td></td>
<td>Wednesday (3)</td>
<td>Leech High School (C)</td>
</tr>
<tr>
<td></td>
<td>Thursday (4)</td>
<td>Brand High School (D)</td>
</tr>
<tr>
<td>May</td>
<td>Friday (5)</td>
<td>Richter High School (E)</td>
</tr>
<tr>
<td></td>
<td>Monday (1)</td>
<td>Hull High School (F)</td>
</tr>
<tr>
<td></td>
<td>Tuesday (2)</td>
<td>Smith Hill High School (A)</td>
</tr>
<tr>
<td></td>
<td>Wednesday (3)</td>
<td>Key High School (B)</td>
</tr>
</tbody>
</table>
To explore the remaining utility with TSK, I also performed some keyword searches for “marijuana” in the keyword search pane, and got a hit (Image 12).

![Image 12: Keyword Search Using Autopsy](image)

4.1.4 Conclusions on Autopsy/TSK

This product is quite feature-rich. Using both logical/physical techniques to dive into allocated/unallocated blocks, there is much that can be recovered forensically from an image. The front-end provides necessary ease-of-use that abstracts most of the unnecessary details away. It also provides for several advanced toolkit features, such as file timelines, that help to piece together both the details and history of the image.

4.2 IPhone Backup Analyzer

IPhone Backup Analyzer (iPBA) is an open-source tool written by Mario Piccinelli specifically written to handle the proprietary and somewhat confusing iOS format. This tool is designed to analyze a backup file (the same backup file produced by an iTunes mobile sync). iPBA through its python front-end & main application loop allows for the user, upon transfer to the host analyzing machine, to simply browse through the contents of the backup folder of the iPhone (or other iOS device, e.g. iPad). Behind the scenes the tool is browsing archives, reading through configuration files, and probing databases. The tool is written in Python, meaning in theory it is cross-platform. However, python on windows using the extension graphical toolkits is not necessarily easy to deploy or setup. The license is very permissive MIT license and allows for both commercial & personal use.
Each file in the backup directory is arranged in a logical order defined by the backup itself, namely: DOMAIN-SUBDOMAIN-PATH-FILE. When browsing particular files each file has the following common elements which help in identifying the finer-grain file information surrounding each item [9]:

- Real name and key (scrambled file name in the backup directory)
- User id, group id and Unix permissions
- Last modify time, last access time, creation time
- Last modification time of the file in the backup directory
- File type (from magic number)
- MD5 hash
- First bytes (in hex) and first chars (in ASCII)

More specific information is shown depending on the file type:

- Plist files are shown (binary plist files are automatically converted in ASCII format)
- Image files are shown
- SQLite files are shown with the list of the tables they contain. By clicking on the tables list the selected table’s content is dumped in the main UI
- Unknown data files are shown as hex/ASCII data

In addition to the file-browser, additional tabular plug-ins are provided for browsing the most interesting data components, namely:

- Contacts browser (with thumbnail)
- Safari bookmarks browser (also shows SMS incoming and outgoing counters)
- SMS browser (messages are arranged in threads as in the iPhone UI)
- Calls history (also shows counters)
- Cell locations (GSM/UMTS cells saved with timestamp and their geographical position)

### 4.2.1 IPhone Backup Analyzer Case Study

I tried downloading iPhone Backup Analyzer to my Windows XP machine since that’s where my wife has historically done backups in the past. Unfortunately, I found the python install to cumbersome. So I abandoned that idea, and instead looked to transfer her backup located at C:\Documents and Settings\(username)\ Application Data\Apple Computer\MobileSync\Backup to my Ubuntu machine. I found out quickly that 5GB was too big for any USB key I had, so I setup a home FTP server and transferred to my Ubuntu laptop.

The install was trivial on Ubuntu. I only had to execute 2 commands, `sudo apt-get install python-tk & sudo apt-get install python-imaging python-imaging-tk`, in order to get the necessary auxiliary packages installed. Then I typed `python main.py` to run the application. After selecting my backup directory I was presented with the main screen. The main-screen presents the forensic analyst with a directory tree (lower-left), general information regarding the image (upper-left), file-browser (middle), and search/database pane (right). (Image 13)
After verifying the MD5 hash (md5sum <file>), with the manifests, I proceeded as any good husband (err, forensic analyst) would do in analyzing the backup. This was a 2 year-old backup so I figured it would be a little difficult putting things in context. I could only imagine the even greater difficulty that a law-enforcement official would have in putting some of these things together. The first thing I looked to was the GPS history (Image 14), which I find very interesting given all the anger over iPhone & other mobile devices saving your locations. In the short-time that my wife owned the iPhone (3 months), it appears she didn’t wander far from the home (limited to Fort Worth-Tarrant County). I decided I would track her location on December 18th, 2010 (a Saturday that she had off) and apparently didn’t wander to far from home. I did discover when trying to understand the results that the iPhone recorded GPS has some error to the tune of approximately 3000ft at most. I took 20 random plots and plotted her locations overlaying a Google map image (image 15). The concentrated plot is the location of our home.
Image 14: GPS Coordinates of Monica on December 18th, 2010

Image 15: Google Maps Location of Monica on 12/18/2010
I also browsed through her Safari bookmarks & browsing history, and didn’t find anything out of the ordinary (Image 16).

However, looking through her text messages the phrase “I really enjoyed hanging out w u the other night.. Thank u” caught my eye. Who is this person that really enjoyed hanging out with my wife? Well cross-correlating the number with her contact information I found that it was her old friend Hailey, who apparently she rekindled her friendship with in late 2010 (Image 17/18).
Image 17: Suspicious SMS message

Image 18: Suspicious Contact
4.2.2 IPhone Backup Analyzer Conclusions

IPhone Backup Analyzer (iPBA) is a pretty neat do-it-yourself forensic tool. You can dive into pretty much any unencrypted iPhone back-up and peel-off an incredible amount of data. The code looks to be still under active development, so I would expect more features. Additionally, one need only understand the code in order to write a plug-in for the tool. The only downside I see is that it doesn’t handle encrypted data. With a cracking front-end I could see this being a powerful tool, commensurate with government application.

4.3 iOS Forensic Tools

It’s worth noting that there are good reviews about this product. It handles both physical and logical acquisition. It also supports pin bypass and decryption of files. Unfortunately it is freely available to government entities only.

4.4 System Tools/Commands

Hexedit is a tool provided on most Linux distributions. It allows the user to view a file as hex or ASCII equivalent. Once in view mode, a user can tab to change between hex/ASCII and search using the / command. This tool is very powerful in locating potential evidence.

Usage: hexedit <file>

The grep command is also very useful in locating evidence quickly. A keyword search can be applied to one file or many files at once.

Usage: grep “keyword” <file>

Strings is a command-line tool that can be used to extract ASCII strings from binary data.

Usage: strings <file>

Scalpel [18] is a command-line tool that can be downloaded freely off the web. It’s useful in examining a file-partition and identifying files which may or may not have been deleted. It carves out what it thinks are files, and throws them in the specified output directory.

Usage: scalpel <file> -o <directory>

log2timeline [19] is a command-line tool that can be downloaded freely off the web. It’s useful in examining file timelines. In essence it acts as a log file parser that produces a body file used to create timelines. This resultant body file can then be imported into other tools for timeline analysis.


Gpart is a command line tool that is useful in ascertaining the primary partition table in case sector 0 is damaged in some way.

Usage: gpart <directory>
5. Summary

There are several techniques & products that can be used to perform a forensic acquisition from a mobile device. It is important to understand there is no 1-size-fits-all solution. It’s possible multiple products need to be used as well as multiple techniques. A forensic examiner should understand the advantages and drawbacks of each product/technique as it relates to the target device in order to acquire forensically sound data that works for their situation. Additionally, it is important that the examiner take necessary precautions to not corrupt the data in any way.
6. References


[19] Log2Timeline Open Source Tool: http://log2timeline.net/