Third Party Application Forensics on Apple Mobile Devices

Alex Levinson
Rochester Institute of Technology
alex.levinson@mail.rit.edu

Bill Stackpole
Rochester Institute of Technology
bill.stackpole@rit.edu

Daryl Johnson
Rochester Institute of Technology
daryl.johnson@rit.edu

Abstract

Forensics on mobile devices is not new. Law enforcement and academia have been performing forensics on mobile devices for the past several years. Forensics on mobile third party applications is new. There have been third party applications on mobile devices before today, but none that provided the number of applications available in the iTunes app store. Mobile forensic software tools predominantly addresses "typical" mobile telephony data - contact information, SMS, and voicemail messages. These tools overlook analysis of information saved in third-party apps. Many third-party applications installed in Apple mobile devices leave forensically relevant artifacts available for inspection. This includes information about user accounts, timestamps, geolocational references, additional contact information, native files, and various media files. This information can be made readily available to law enforcement through simple and easy-to-use techniques.

1. Introduction

The operative word when describing mobile devices is “mobile”. Individuals carry cellular phones and other mobile devices with them everywhere. Forensic examiners have learned that information from such devices can be invaluable to an investigation. The data stored about the user can provide information about with whom they communicate and where they have traveled, all tied to a common time source (the cellular provider’s system clock.) So-called “smart phones” have expanded the amount of information stored about a user to include email history, location information stored by the device, usernames, passwords, wireless access point associations and other useful information. [1] With the introduction of an application marketplace (commonly referred to as an “app store”), the applications stored on the device have increasingly changed from being completely under the control of the device provider to being defined by the user.

1.1. Apple Devices

With the introduction of the iPhone, Apple Computer has created a mobile handheld platform that allows users to install and configure a wide variety of applications via their “app store”. The iPad device, introduced in April 2010, runs most iPhone apps in full functionality, as well as some that have been modified specifically for use with this larger format device. Users select applications of their choice and install them on the device. The application is downloaded to the device from Apple’s servers and installed. The application can now be launched by the user. The application can store data about the user that customizes the app for their use or stores information about how and when they interact with the app. Apps are typically backed up to the personal computer of the user whenever the device is synced as well.

Applications can be written by anyone with sufficient programming expertise after they agree to the terms prescribed in the Apple Developers License. Apple closely regulates applications submitted for sale in the app store. Applications can be denied inclusion in the app store based on the terms outlined in the Apple Developer License. For example, apps whose sole purpose is to display prurient images are likely to be declined. There does not, however, appear to be a standard to which application developers must adhere with respect to how or where applications store information, whether generated by the application or provided by the user. Applications request information from a user in order for the application to be customized with personal preferences of the user. Requested information can also allow applications to store credentials about the user to facilitate connections to other servers.

Apple’s Development SDK contains a number of programming classes to allow a developer to store application data locally on the phone. Using the programming standards provided by Apple, third party application data is typically stored in plaintext format. When users interact with their “apps”, the information provided by the user is stored in the device and can be made accessible to a forensic examiner.

1.2. Application & Platform Growth

The number of applications available for the Apple mobile device platform has grown exponentially since the App Store’s inception in May of 2008. There are currently over 200,000 active applications in the app store. [2] According to Steve Jobs presentation during the iPad announcement in April 2010, the number of iPhone OS-based devices exceeded eighty-five million. [3] This included both iPhone and iPod touch devices. This expansive growth has given Apple single platform dominance in the mobile application market. [4] As of June 1 2010, two million iPad devices had also been sold. The price point for the entry-level iPhone and iPod touch devices has been dropping, making the devices available to more users than ever. The Apple mobile device platform is popular and will continue to grow as a platform for third party applications. This growth will continue to produce applications that store forensically rich data.

1.3. Proliferation & Constant Use

Mobile phones and other mobile devices are becoming ever-present as the main technology platform in cultures around the world. [5] More people are obtaining and using mobile phones than ever before. Many are using them in place of landline phones. As part of an always-on, always-connected society, mobile computing is becoming ingrained at an earlier age. According to the Pew Internet and American Life project, 71% of teens ages 10-17 own a cell phone.[6] In the college environment, many students use smart phones. One would be hard-pressed to find a student who does not use some sort of mobile device for communication. The barrier to entry for smart phone use is dropping as device subsidies from cellular carriers serve to amortize the cost of the phone over time.

In addition to being communication devices with contact information and history, cell phones keep accurate track of time. [7] Apple mobile devices can also capture photos embedded with location information about the images in the Exchangeable Image File Format (EXIF) metadata. [8][9] These devices can also manage schedules via the built-in calendar application, update social networking websites such as Facebook, MySpace and LinkedIn. Other functions of this mobile platform includes the ability to take notes, read books and periodicals, manage shopping lists, email, instant messages, and perform many other tasks. There is much more information on these devices than simply telephony, SMS, and pictures. Users of Apple mobile devices tend to store everything that is important to their day-to-day lives on these ubiquitous, convenient, and easily-carried devices.

1.4. Forensic Relevance

Mobile forensics continues to garner the support of analysts around the world. Considering the amount of communication facilitated by the use of mobile devices, the mobile platform creates an effective way to correlate data and provide a forensic timeline surrounding their usage. With the introduction of smart phones, especially Apple’s mobile device platform, third party applications have become widely used. While many forensic tools are available to interpret typical mobile telephony data on an Apple mobile device, a commercial tool has not yet been developed to extract relevant data from all third party applications.

Apple mobile devices can be used to view, process and store general purpose documents in their native format. These files, including doc, pdf, and pages, are likely to provide relevant and timely information to the forensic analyst. [10] Using techniques developed by the authors of this paper, analysts can extract more information. They may be able to use that information to accurately reconstruct a forensically-relevant timeline of activitiesthat may have been performed on the Apple mobile device.

In addition to the data stored by third-party applications, simply reviewing the condition of the device and the apps installed on it may provide insight into the owner of the device. For example, if a device is “jailbroken” and contains network evaluation tools, the mere presence of such tools can shed light on the technical capability of the individual from whom the device was obtained.

2. Methodology

2.1. Data Partitions

Apple mobile devices use two data partitions in which to store information. The System partition contains many underlying components of the operating system as well as all executables. The User Data partition contains configuration information for both the operating system and all applications. The 30-pin connector cable (aka “sync cable”) provides direct access to the data partition ONLY; the system partition is not available for reading via the sync cable in the phone’s default configuration. Devices that have been modified, or ”jailbroken”, violate this direct access limitation and may be able to access data, but is vulnerable to data corruption. [10]

2.1.1. Obtaining the User Data Partition. There are some commercial tools that provide access to data stored in the User Data Partition. These include Lantern by Katana Forensics and Oxygen by Oxygen Software Company. Given the popularity of Apple’s platform, other commercial tools are likely in development. Such tools will leverage the ability to access the content stored in the User Data partition. Commercial tools are slowly providing access to information stored by select third-party application providers. Oxygen, for example, has recently released an update to their tool to allow access to information about the Skype application and WiFi connections.

2.1.2. Examining Files Inside the User Data Partition. Data on the User Data Partition is stored predominantly in *.plist and SQLite database formats. A plist file is a properties file that contains a dictionary of keys paired with a value. Apple uses plists properties in both OS X and iPhone OS operating systems. Plist files come in binary and XML format. Apple provides a command line utility, (*plutil*, on Mac OSX 10.6) for converting plist files between XML and binary, as well as a utility for viewing both binary-based and XML-based plist files. SQLite databases can also be interrogated through a command line utility, (*sqlite3*, built into OS X 10.6).

Directory	Description
/var	Encrypted Password Storage
/private	3rd Party Application Data
/Library	Telephony & iPhone Built-in Data
/Media	Pictures & Videos from Camera
/iTunes_Control	Media synced from iTunes

Figure 1: Table of the top level directories inside the User Data Partition.

There are five directories inside the User Data Partition. The directory names are: */var*, */private*, */Library*, */Media*, and */iTunes_Control*. Each directory contains a different set of information. The */var* directory contains a SQLite database with important password information stored in it. This data is encrypted inside the database. The */private* directory contains all third party application data. The */Media* directory contains photos captured on the phone while */iTunes_Control* contains all the information regarding the iPod library. Finally, the */Library* directory contains most of the configurable phone data: SMS database, contacts database, plist configurations, etc. Current tools know where to find specific files that correspond to relevant device data; the locations are standardized across the User Data Partition. There is no such standard for third party application data. For this reason, third party application data is being missed by the current crop of commercial mobile device forensic tools. Both built-in application data and third party application data can prove invaluable to a forensic analyst, provided that the relevant data can be extracted from third-party applications.

2.2. Data From Built-in Applications

Commercial tools typically focus on interrogation of built-in application data. Built-in applications store data in plist and SQLite database formats. Media such as music, movies, and podcasts synced to the device through iTunes are stored in the */iTunes_Control* directory. The static location of these directories provides commercial tools the ability to find the same types of data in the same relative locations across multiple Apple mobile devices, regardless of how the user has configured the device.

```
<?xml     version="1.0"     encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple// DTD PLIST
1.0//EN"     "http://     www.apple.com/DTDs/
PropertyList-1.0.dtd"> <plist version="1.0">
    <dict>
      <key>last-logged-in</key>
        <date>2010-06-15T07:00:00Z</ date>
      <key>password</key>
        <string>fake-pass</string>
      <key>username</key>
        <string>fake-user</string>
    </dict>
</plist>
```

Figure 2: Example of the XML data that can be viewed in a plist file.

2.2.1. Camera Data. Images captured by the iPhone camera (and video on compatible models) include EXIF data tags. [6][7] This data is stored on the phone inside the */Media/100APPLE/* directory and may be included when copies of an image are transferred to other media or locations. Once extracted, the EXIF data can be examined using tools such as Preview on Mac OS X to find the geographic coordinates embedded at the time the picture was taken. These pictures can be found in the same location for every

Apple mobile device with a compatible camera. This data can reveal the time and location the device was used to capture the image.

2.2.2. WiFi Data. WiFi association information may also be used to relate date, time and geo-location. WiFi association history is stored in a plist file. In conjunction with DHCP or other data, these associations can be used to place a device at a given location as well as to link traffic (email, web, or otherwise) to a given IP address or Access Point. Association information can be found in the file */Library/Preferences/SystemConfiguration…*
…/com.apple/wifi.plist. Apple mobile devices retain this data in order to create a list of known WiFi networks with which the device has associated. This allows a device to auto- associate with access points to which it has previously connected. While useful to device owners, the WiFi data is readily and accessible to a forensic analyst.

2.2.3. Maps Data. The Google Maps application has the capability to store bookmarks, recent map searches, driving directions, and user contact address locations. This application also stores geo-location data about the last coordinate found. This data can help place a device at a specific location or substantiate interest in a geographic locale.

2.2.4. Device Dictionary. All words typed into the device via the virtual keyboard are archived in the */Library/Keyboard/en_US-dynamic-text.dat* file. Examination of this file shows all keyboard entries of the user. This can include things the user has said that has since been erased.

2.2.5. Clock Data. There are three different types of time standards used by Apple mobile devices. The first is the Unix epoch time. This standard is widely used throughout the Linux/Unix computing world. The second is Apple's own implementation, AbsoluteTime. AbsoluteTime is the number of seconds since January 1st, 2001. The third timestamp that can be found is standard UTC dates. UTC timestamp artifacts are typically used in third party applications.

2.2.6. Other Applications. Other applications also store information about user activity. The browser on Apple mobile devices, Safari, stores web history and bookmarks on the phone. Mobile Safari on the Apple mobile devices fully support the new web standard, HyperText Markup Language version 5 (HTML5). HTML5 provides standards by which web developers can create databases of information and store them locally on the device. Websites such as Google Mail are incorporating this into the versions for Apple mobile devices in order to provide users with offline functionality of websites. Bookmarks, histories, and even HTML5 local databases can be accessed through interrogation of files inside the */Library* directory. The notes application is also available for viewing in this directory and contains timestamps for the notes saved. Many artifacts present inside the built-in application data are of interest to the forensic analyst. The amount of data that can be obtained from these can be valuable to an investigation.

2.3. Data from 3rd Party Applications

While retrieving relevant data from built-in applications is important, there is also forensically-rich data stored by third party applications. Third party applications often contain social networking constructs such as social messaging, contacts, or current and past location. Applications store varying amounts of data.

Because there is no standard on data storage (other than the plaintext methods provided by Apple), the developer is left free to store data as they see fit. While different applications may hold similar types of information, they often store it in different locations and formats. Viewing the contents of the */private* directory can yield useful information to the examiner.

The Apple mobile operating system, iOS, does not currently support background execution of 3rd party applications (At the time of this writing). With the introduction of iOS v4, background third party application execution is enabled, but concurrent execution is limited. Typically, multiple applications do not run concurrently. In addition, applications are executed within a "sandbox" environment. This prevents any application from directly interacting with data stored by other applications. There are methods that allow a third party application to launch built-in phone functions, such as generating an email. Third party applications, however, cannot access data stored by other third party applications. Similar to the Unix *chroot* function, sandboxing essentially "jails" an application and prevents it from accessing any place other than an isolated part of the filesystem structure assigned to that application.

Third party applications are stored under the parent directory */private/var/mobile/Applications/* in the User Data Partition. Each is assigned a value generated by the operating system when the application is installed. Inside each directory, there are at least two documents. One is an image in JPEG format named "iTunesArtwork". This image is presented to the user as an on- screen icon to identify the app in the App Store Application – the built-in application used to purchase and download third party applications. The second file present in all third party application directories is *iTunesMetaData.plis*t which contains information about the application itself. Inside *iTunesMetaData.plist*, tags useful in identifying the application can be found. Some of the tags include iTunes IDs of the publishers, purchaser, their corresponding names, purchase date, purchase cost, and software version.

When interrogating third party application directories, there are six items an examiner might find forensically relevant to an investigation. Each of the following six items can provide insight into past use of the device. Together, they can create a forensically rich portrait of both the user and their activities.

2.3.1. User Account Information. Some applications require authentication and may store a username and/or password. This will depend on what methods are employed to authenticate the user. Many of these applications have their own *[app-folder-name]/Library/Preferences/* subdirectory with a plist containing this information. This storage location appears to be a convention followed by most developers, although not an explicit standard from Apple. Authentication information is commonly found in this plist file. Instant Messaging applications are one example of a program where authentication information is likely to be available to an examiner. If a password is not explicitly stored, authentication information may be accessible through interrogation of a password hash or by exploiting persistent session cookies. These could be used to obtain access to other data stores related to the device user.

2.3.2. Timestamps. Many applications also store timestamps. Applications often use timestamp data when updates are required at some time interval. Timestamps are also present throughout many application preference plist files. Such data can be useful for providing a time reference for device usage. Time stamps can be used to correlate associated phone usage with an application. Remember that applications are not allowed to run in the background; the app had to be running when the timestamp was created.

2.3.3. Geolocational References. Apple mobile devices can use any of three different technologies to assess geographic location of the device. The iPhone and iPad 3G models contain GPS chips. The cellular towers for these devices are also integrated into the geo-location algorithms. Apple mobile devices also take advantage of Skyhook Wireless Technologies to provide geographic positioning based on fixed wireless access point locations.

Apple provides an API that allows developers to access this data. Many applications use geo-location information to provide data related to a phone's current location. The data is often stored inside application preferences similar to the way timestamps and authentication information are stored. Analysis of the geolocational data and other additional information can substantiate that a device was at a given location. An examiner can use timestamps stored in third-party application data files to corroborate both time and position.

2.3.4. Contacts. Commercial tools usually recover the contacts built into the "phonebook" within the phone. Social networking applications can store a separate set of unique contacts. Examples might include Skype or Facebook. The Skype application on the iPhone keeps a proprietary file with all the contacts in it named *user256.dbb*. A string search of the file will reveal user information. Not every application stores contact information in the same manner or place, but the fact that the data exists may allow an examiner to access it.

2.3.5. Native Files. Apple mobile devices allow document viewing and editing. While more common with the Apple iPad, document viewing and editing is also possible on the iPhone and iPod touch as well. Document editing applications such as Apple's Pages, have been ported from the desktop application to the iPhone OS. Inside the application directory for Pages, any document that has been loaded onto the device can be recovered. Newly created documents reside in a different directory, *ApplicationSupport/*, and are a combination of image files and SQLite databases, extractable through SQLite database analysis.

Pages is just one of the many applications on the iPhone OS that can be used to view, edit, or generate documents. Other applications may include code editors, PDF readers, or other document management applications. Whatever the application, files read can be stored locally and should be recoverable within the application directory.

2.3.6. Media – Books, Music, Etc. There are several new ways consumers are interacting with media on Apple devices. eBooks may now be viewed on mobile devices, movies and music can be streamed, and video calls may be possible over WiFi or cellular data connections. This information can help to develop a profile of the device user.

3. Applied Analysis

Within the last few years, the world of social networking has exploded. Many of the third party applications demonstrated here are used as social networking platforms. In contrast to traditional text messages and phone calls, social networking provides an ecosystem rich with participation and opulent data about users' activities. Recently, these third party social networks have pushed their way onto mobile devices intended for on-the-go use.

As mobile devices continue to store more than just telephony data, forensics analysts need to stay current on what data can be stored as well as how this new data can be accessed in an investigation. To better understand the importance of this data, the authors have laid out a mock scenario where third party application data provides key evidence and leads.

3.1. Case Background

The subject of the investigation is a Mr. John Doe. John works as a sales rep for a prominent company in the area and is married to Mrs. Elaine Doe. They both are Apple mobile device users. The day is Thursday and Elaine expected John to return home late. He expressed the need to stay late at work to her. By midnight, he had not returned and she became worried. Using an Apple product, MobileMe BackToMyiPhone, Elaine was able to determine the location of John's phone. She located the device in a Walmart parking lot. At this point, she called the police to investigate and locate her husband.

She informed the investigators that the communication she had with John was around 4:30pm over Facebook chat. She had been at home on her laptop while he was using his phone. This is where he communicated to her he would be home late - sometime around 8:00pm. She was otherwise unaware of any activities he had scheduled. The investigators promptly imaged the phone on site and began their analysis.

3.2. Forensics Image Overview

After the phone was imaged, the forensic analyst began sifting through the phone data to bookmark anything that might be important. He noticed John had a number of third party applications installed on his phone, including:

- Twitter *(broadcast / status service)*
- Facebook *(friend networking)*
- Skype *(communication)*
- FourSquare *(geo-locational social gaming)*
- BrightKite *(geo-locational / group messaging)*
- Where.com *(geo-locational commerce)*
- iBooks *(PDF/eBook viewer)*
- Yelp *(geo-locational restaurant locator)*

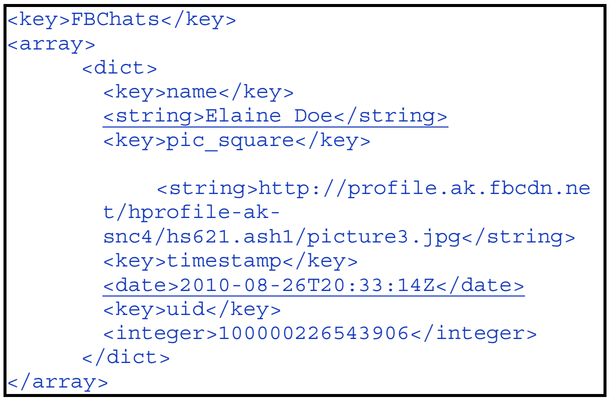
As will be seen in the investigation, these applications include information and forensic artifacts that will be shown to have direct relevance to solving the case. The investigator checked the standard locations in the phone's built in application data. John had placed no calls and sent no text messages that yielded information helpful to the situation.

The best practice for dealing with multiple data sources is for the analyst to build a forensic timeline of events. As he began interrogating the third party application data he was able to insert artifacts from each application into a forensic timeline to shed light on John's possible location.

3.3. Timeline Analysis

a.04:33PM - Facebook

There was data that showed recent chat's John had on through Facebook chat on his iPhone. We can corroborate that he did use Facebook to chat with his wife.



```
<key>FBChats</key>
<array>
      <dict>
        <key>name</key>
        <string>Elaine Doe</string>
        <key>pic_square</key>

             <string>http://profile.ak.fbcdn.ne
        t/hprofile-ak-
        snc4/hs621.ash1/picture3.jpg</string>
        <key>timestamp</key>
        <date>2010-08-26T20:33:14Z</date>
        <key>uid</key>
        <integer>100000226543906</integer>
      </dict>
</array>
```

Figure 3: com.facebook.Facebook.plist - Recent chat metadata including date and timestamp

Curiously, there was a reference to another Facebook account under a different name, "James Notreal". After going to the user's profile, it had the same picture as John's Facebook account in it. This could potentially be a code name for John.

```
<key>FBUserInfo</key>
<dict>
    <key>1335450346</key>
    <dict>
      <key>name</key>
      <string>John Doe</string>
      <key>pic_square</key>

      <string>http://profile.ak.fbcdn.net/v229
    /1319/7/picture1.jpg</string>
    </dict>
    <key>513209787</key>
    <dict>
      <key>name</key>
      <string>James Notreal</string>
      <key>pic</key>

      <string>http://profile.ak.fbcdn.net/hpro
    file-ak-
    snc4/hs629.ash1/picture2.jpg</string>
    </dict>
</dict>
```

Figure 4: com.facebook.Facebook.plist - Listing accounts that have accessed Facebook from this device.

b. Unknown - iBooks

A file titled Party-Invitation.pdf was found in the iBooks application directory. There is a record inside the iBooks configuration that shows the last document viewed.

```
<key>BKMainViewControllerLastBook</key>
<string>71BBF62738191026BBDB3EEB75AA7F91
</string>
```

Figure 5: com.apple.iBooks.plist - Last Book viewed in iBooks

To determine the file, the investigator opened a file inside the iBooks application directory titled, *iBooks_v1.1.1_07142010_3_8A400.sqlite*. Inside the ZBKBOOKINFO table, there was a direct reference from the <string> above to the file, Party-Invitation.pdf, confirming that this file had been viewed.

c. 05:34PM - Yelp

A search record was parsed from within the Yelp application data. The search was for a bar called “The Fox and the Hound” from Thursday at 5:34pm. This search data was determined by cross referencing various SQLite data (*business.sqlite, user_defaults.sqlite*) with the Yelp configuration plist. The investigator then was able to corroborate John’s intention of going to the party.

d. 06:05PM - FourSquare

The investigator found a record of a FourSquare “check-in” by the suspicious other Facebook user, James Notreal. (A Check-in is an action a user takes that “beacons” his location and makes it viewable to other FourSquare users. He can then see who’s geographically nearby or possibly at his current location.) Here’s the record of his check-in:

```
<string>2956906</string>...
<string>James</string>...
<string>Notreal</string>...
<string>173959</string>...
<string>Fox and Hound</string>
<string>14490 Lowes Way</string>...
<string>Carmel</string>...
<string>IN</string>...
<string>46032</string>...
<string>146th St</string>...
<string>3178440075</string>...
<real>39.997795000000004</real>...
<real>-86.122599199999996</real>...
```

Figure 6: GroupedCheckins2.archive (pList) - Recent chat metadata

e. 07:20PM - Twitter

Next in the timeline, the investigator found a tweet from the user, @FriendOfJohns. This tweet stated, “Beer Pong Tournament at Kona Bar and Grill!”.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST
       1.0//EN"
       "http://www.apple.com/DTDs/PropertyList
       -1.0.dtd">
<plist version="1.0">...
<real>304452898.125657</real>...
<string>Beer Pong Tournament at Kona Bar and
       Grill!</string>...
<string>FriendOfJohns</string>...
</plist>
```

Figure 7: 45049D6F-5B2E-48F1-B029-3C0C02D53D7B - snippet of XML from a binary plist inside the twitter application directory.

f. 08:42PM - BrightKite

Inspection of the BrightKite application was promising. It provided a username and password in plaintext, located in the *$APP_ROOT/Library/Preferences/com.brightkite.Brightkite.plist*. It also contained information about the last group text (text messaging through BrightKite with multiple participants) that John had sent. The BrightKite account was under “jamesnotreal”.

```
<key>username</key>...
<string>jamesnotreal</string>...
<key>password</key>...
<string>jamespassword</string>...
```

Figure 8: com.brightkite.Brightkite.plist - User account information in plaintext - this could be

used to log onto the web.

```
<string>585***8526</string>...
<string>jamesnotreal</string>...
<string>Allice, its james - come to the kona
       bar and grill</string>...
```

Figure 9: conversations.dat - An artifact containing a record of a text message. This was sent from jamesnotreal to a phone with the area code of 585.

g. 10:53PM - Skype

While browsing the data for keywords - specifically timestamps - the investigator found the timestamp "11pm" inside a binary data file in Skype. This was a message between two users, one being jamesnotreal, the other username being "maraudintedybear", aliased, "Allice Fake". While the message indicates that it is unread, it may have been viewed via a push notification without the user having directly checked the application.

```
<?xml version="1.0"?>
<config        version="1.0"        serial="60"
       timestamp="1282877580">...
```

Figure 10: config.xml - Timestamps the last time the Skype application was used.

```
Ãù§ì
       ∏#maraudintedybear/$jamesnotreal;4fd377
       4adb2b973d32068076c431aafed.dat≥Ü–„   µ
       ¬ÈÀ„ A      · ≈π /°
maraudintedybearâÃjamesnotreal
maraudintedybearÿAllice Fake | 11pm, my place
'jamesnotreal maraudintedybearΩ∏è¿„
¿maraudintedybear-»jamesnotreal
       maraudintedybear
```

Figure 11: chat512.dbb - Recent chat metadata showing contact between jamesnotreal and Allice. This file was in binary format so a string search had to be performed.

At this point, it's suspected that John left the Kona Bar and Grill and travelled to Allice Fake's location.

h. 11:41PM - Where.com

The last artifact the investigator came across was a plist inside the Where.com application. There was both user account information and geolocational information present. (see figures 11 and 12)

```
<key>loggedInUser</key>
<string>{"user":{"demographics":{"gender":"U",
       "birthyear":"(null)"},"password":"hello
       pass","username":"johndoegmailcom","aut
       h_token":"fd0cc08c-e330-48a9-b7a0-
       3fa73c19d628","email":"johndoe@gmail.co
       m","settings":{"contact_mdn":"317382777
       7","email_allowed":false,"sms_allowed":
       false}}}</string>...
```

Figure 12: com.ulocate.where.plist - User account information including plaintext username and password for the Where.com application.

```
<date>2010-08-27T03:41:26Z</date>...
<key>lastLocation</key>
<string>{"location"{"city":"Carmel","place_nam
       e":null,"state_long":"Indiana","postal"
       :"46033","street":"14472        Jeremy
       Dr","state":"IN","email":null,"lat":39.
       998437,"heading":-
       86.09363496,"accuracy":1499,"building":
       "686","distance_from_some_location":nul
       l,"phone":null,"lng":-
       86.117237,"geocoded":"YES","country":"U
       S","speed":-1}}
</string>...
```

Figure 13: com.ulocate.where.plist - Geolocational information down to the exact adress about where the application was being used from.

3.4. Timeline Conclusion

From the collected data, a rich forensic timeline has been built surrounding John Doe's activities.

Time EDT	Data	Location	Application
04:33PM	chat record with wife	office	Facebook
04:33PM	alternate account discovered	office	Facebook
Unknown	party invitation	unknown	iBooks
05:34PM	party location search	unknown	Yelp
06:05PM	social networking check-in	Fox And Hound Bar (party location)	FourSquare
07:20PM	post by friend inviting to other bar	Fox And Hound Bar (party location)	Twitter
08:42PM	text message inviting "Allice" to current location	Kona Bar and Grill	BrightKite
10:53PM	message from Allice with a meeting time of 11PM at her place	Unknown	Skype
11:41PM	Last Location	14472 Jeremy Dr, Carmel, IN	Where.com

Figure 14: Timeline summary.

Officers were dispatched to the last location in the forensic timeline. At this address, they found two persons - John Doe and Allice Fake. The home was that of Allice. Using the data contained within the third party applications on John's phone not only lead investigators directly to his location, but also helped provide a clear picture of the events leading up to that point.

4. Continuing Research

As the application market continues to grow, the demand for utilities capable of extracting forensically interesting data from mobile devices will also grow. The examination of forensic images could be automated in such a way that any forensically relevant data is returned to a log whose contents could easily be reviewed. This could happen through the use of regular expression pattern matching against key terms from a static dictionary and a dynamic device-specific dictionary. The static dictionary could contain forensically relevant terms such as "password" and "timestamp", while the dynamic dictionary could be comprised of information related to data extracted from the device. For example, every first and last name found in the contact list would be added to the dictionary. This could help provide an accurate map of contact names available through third party applications.

5. Conclusion

Third-party applications on the Apple Mobile platform contain a significant amount of data that can provide relevant information to a forensic analyst. Information provided to the device by the user through their interaction with applications is typically stored in plaintext format and can be extracted from the User Data partition of the mobile device. Information can include authentication credentials, time-stamps, and geolocational references that can help to place a device in a location at a specific time. This data may be used to extend an investigation on other platforms should user credentials be mirrored or otherwise duplicated. Data available from third-party apps can be obtained from a forensic image of the device. Third-party application data may also be available through interrogation of backups stored on the machine with which the device has been synced. The techniques to obtain this information are simple. Integrating them into various mobile device forensic platforms can provide significant benefit to the forensic community as well as to law enforcement.

5. References

[1] Ayers, Richard. "Mobile Device Forensics - Tool Testing". National Institute of Standards and Technology May 6 2009: 1-23.

[2] http://148apps.biz/app-store-metrics/ [3] Jobs, Steve – Apple keynote, April 2010 - http:// www.youtube.com/watch?v=lTNbKCAFHJo

[4] http://www.businessweek.com/news/2010-06-13/nokia- loses-battle-for-apps-as-iphone-android-snare-developers.html

[5] Marsico, C., Rogers, M. "iPod Forensics",International Journal of Digital Evidence, Fall 2005, 4-2.

[6] Lenhart, et al, Teens and Mobile Phones, http:// www.pewinternet.org/Reports/2010/Teens-and-Mobile-Phones.aspx

[7] Martinez, Javier. "The Roadmap to Mobile Forensics." Speech. Mobile Forensics World 2008 Conference. Chicago. 2008

[8] Toyama, K., Logan, R., and Roseway, A. 2003. Geographic location tags on digital images. In Proceedings of the Eleventh ACM international Conference on Multimedia (Berkeley, CA, USA, November 02 - 08, 2003). MULTIMEDIA '03. ACM, New York, NY, 156-166. DOI= http://doi.acm.org/10.1145/957013.957046

[9] Standard of the Camera and Imaging Products Association. CIPA. Exchangeable image file format for digital still camera: Exif Version 2.3. CIPA. 2010

[10] Jansen, Wayne. "Guidelines on Cellphone Forensics". National Institute of Standards and Technology May 2007: 1-104.

[11] Morrissey, Sean. Mobile Forensic Analyst, Federal Forensics Lab. Personal interview. 10 Jun. 2010 Link: http:// www.amazon.com/Sean-Morrissey/e/B002Z06C3M