Examining Social Media on Mobile Devices

What data to look for, where to find it, and the dangers lost data can pose

By: Kevin Swartz

5/11/2012

Abstract

Over the past seven years, Facebook.com has become one of the most popular websites on the internet with over 800 million users. [1] Its unique combination of attributes as a social media website such as photo and video sharing, 'wall posting', and full person to person chats have made it a viable communication medium for individuals and businesses alike. Other social media websites such as Twitter and Skype have seen similar increases in use over the past few years. While these websites increase the ability for users to communicate to their target audiences, the security of the information sent over these sites hasn't been nearly as vetted. As a result, there may be a large amount of confidential data 'left behind' on devices that connect to these sites.

This study is a forensic analysis of social media applets on mobile devices. It is a focus on the artifacts and data left behind by these applets, such as user names and passwords. It will also discuss the dangers that these security fallacies can present to a business, while also detailing ways to prevent against insecure applications and data breaches moving forward.

This study will focus on two mobile platforms in particular: Google's Android and Apple's iOS. It will discuss what data is recoverable from devices running those platforms and will detail ways to prevent against data loss on the platforms as well. It will disclose the step-by-step processes that attackers can use to obtain critical data. It will show that personal and highly confidential data can be found in the clear on mobile devices and that any implied security on mobile devices should be vetted before being trusted. It will show that social media applications can contain a wealth of information which can be used against unsuspecting businesses and individuals if they are not careful with their data.

Table of Contents

Introduction

In 2004, a website was created called “thefacebook” [2] . It was created as one of the first of a new breed of social websites where users could input their most personal data: names, addresses, emails, phone numbers, and more. Since its inception in 2004 and subsequent name change to “Facebook”, the website has exploded to over 800 million users and is one of the most popular websites on the internet [1]. While it was first created as a website to connect individuals, Facebook now features full webpages designed by businesses, small and large. It allows full text conversations in pop-out windows, personalized advertising, and friends’ lists. It allows for photo albums, video uploads, and more. Over the past 8 years, it has become a ‘one-stop shop’ for people all over the world to socialize, advertise, and connect.

Businesses have capitalized on its popularity in unprecedented ways. Companies on television and radio persuade consumers to “‘Like’ us on Facebook” and “Check out our Facebook page at (www.companyname.com).” While this has proven to be a fantastic form of marketing for businesses [3], it has also presented new challenges for information security. Businesses have Facebook profiles with data about consumers, conversations with interested customers, and may have internal pictures that are not meant to be seen by the public. Facebook’s other features, such as instant chat and Facebook messages, have developed into new forms of communication for internal and external use as well.

Facebook is just one example of many. Other websites, such as LinkedIn, Twitter, and email sites such as Gmail and Hotmail provide many of the same services that Facebook provides. These sites’ popularity has coincided with the explosive rise of smartphones and tablets. According to the Nielsen rating company, 46% of all mobile phone owners have smartphones. That trend is increasing, as Nielsen also found that 60% of people that purchased a phone from October to December of 2011 purchased a smartphone. Of the smartphones that are purchased, phones running on an Android or iOS platform are dominating the market. 46.3% of smartphones are Androids while 30% are iPhones. [4]

These smartphones are more than just devices that can be used to make phone calls. Today’s smartphones are fully functioning computers capable of accessing and storing data far greater than in years past. These new phones can access the internet, stream videos, and even allow for video chat. They can store videos and songs while also running games and hosting apps. To be able to run these features and keep up with the ever changing needs of consumers, high powered processors, large amounts of RAM, as well as memory and CPU caches have been developed and implemented in modern smartphones.

The inevitable explosive rise and subsequent combination of these two communication mediums has created security issues for individuals and businesses alike. The CIA triad, an information security model, describes the idea that there are three main aspects to information security: confidentiality, integrity, and availability [5]. This model applies to mobile phones and their applications as well as computers. For example, many users do not want to input their login information every time they go to Facebook.com, so they instruct their computer to remember their user ID and password. They may update their phone number on the website or post photos of a family trip last weekend. Much of this data gets stored on their phone, even after being deleted. These features add availability, however unless the programmers or users take the correct steps this data can be retrieved by someone with the right tools and correct know-how, decreasing data confidentiality. While this data may be insignificant to an individual if lost (for example, data pertaining to an individual’s birthdate or a conversation about what someone did last Saturday), it could also be highly valuable data such as a

social security number or credit card information. Without being aware of the information that is gathered on computers, specifically mobile devices, businesses and individuals can be creating serious security issues for themselves and their business.

This study is going to discover and detail many of the artifacts left behind by social media websites. It is going to focus on social media applications Facebook, Skype, and Twitter as well as other applications such as Gmail and Flickr on Android and iPad devices. The study is going to focus on finding data that is vulnerable to forensic attacks. It will detail the processes and programs that can be utilized to conduct these attacks. It will also describe different ways to prevent this data loss from occurring. This will include a discussion of programs, applications, and security-specific companies that can be utilized to protect and effectively wipe data. It will conclude the research by recapping the objectives, discussing the results of the artifacts found on the devices, and detailing how this information can be used against the device owners.

The Rise of Mobile Computing

Traditionally, security and forensics have lagged behind innovation in an industry. Nowhere is this more apparent than in mobile forensics. While mobile phones have held meaningful forensic data since their inception as a major communication medium, the ability to extract that data has struggled to keep pace. While mobile devices have been increasing in popularity over the past 15 years, many of the books that have been written on the subject of mobile forensics have appeared on shelves over just the past 5 to 6 years. Like most cutting edge technologies, many of these books are now out-of-date with the current technology. Modern studies in mobile forensics have focused on the ability to obtain data from the most popular platforms, such as Android devices and iPhones. Others focus on more of a well-rounded approach to mobile forensics, highlighting proper forensic procedures and common technical issues. Both types of books were utilized for this study.

Andrew Hoog's book on Android forensics is a comprehensive evaluation of the Android operating system as well as proper forensic techniques. [6] Andrew builds on years of experience as the Chief Investigative Officer for viaForensics, a leading computer and mobile phone forensics company based out of Oak Park, Illinois. The book is a well-structured resource for new examiners and experienced forensic technicians alike. It discusses the origins and structure of the Android system as well as open source tools to assist forensic examiners in determining the location of artifacts left on Android devices. The book also dissects multiple commercial forensics products that work with Android devices that are available for use today. Similarly, the book on iOS forensics written by both Andrew Hoog and Katie Strzempka [7] provides a wealth of detail into the current state of forensics on Apple devices. Both books, along with Sean Morrissey's iOS Forensic Analysis [8], are key additions to any mobile forensic analysts' library.

The study of the rise of social media and mobile phones has been well-documented in both the news and current literature. Companies such as Nielsen have a vested interest in the world of mobile phones and, as such, have written many articles on the topic, such as their article "More US Consumers Choosing Smartphones as Apple Closes the Gap on Android" [4] which details the recent buying patterns of consumers. Mary Smith and Chris Treadaway go into a detailed discussion of the effect of Facebook on American business in their book "Facebook Marketing: An hour a day" [3]. Among others, these books provide valuable insight into the influx of social media websites and platforms, their rise alongside mobile computing, and the effect that they have on consumers and businesses alike. They have proved vital during this study for a full understanding of the ever changing mobile device and social media industries and are highly recommended for anyone studying this aspect of forensics.

The Purpose and Goals of a Mobile Forensic Examination

There are multiple steps involved in a forensic analysis, including the preparation before the examination as well as the proper handling of devices and evidence afterward. According to Adrian Palmer in his report "Computer Forensics: The Six Steps" [9], there are six main steps in a computer forensic procedure. They are:

1) Consultancy
2) Data Preservation
3) Data Collection
4) Data Recovery
5) Computer Forensic Analysis
6) Expert Reports & Testimony

Palmer describes consultancy as discussing strategies for collecting, analyzing, and processing data. As discussed, mobile forensics is a new and constantly evolving industry. As such, while there are many unique resources available to consult, this study focused on recently written books and other materials as sources. As Palmer describes in his report, the goal of consultancy is to understand the properties of the device to be analyzed. This research should include file system structures, user data and system information. Step 1, therefore, is to determine what is to be found, where it may be found, and why it is being sought. [9]

The second main step in a forensic analysis is data preservation. Palmer discusses four main points that a forensic expert should ensure in this stage: "First, potential evidence is not damaged. Second, viruses are not introduced. Third, extracted data is protected from mechanical or electromagnetic damage. And lastly, a proper chain of custody is maintained throughout the process." [9] Each of these aspects of preservation plays a key role in mobile forensics. For example, most mobile devices are found while on battery power alone. In this situation, the examiner has only a limited amount of time to obtain critical data. If the phone battery dies before an examination can be conducted, volatile data critical to an investigation can be lost.

A proper chain of custody involves making as few modifications to a device as possible. A truly sound forensic examination would include obtaining sought data without making a single modification to the target device. This includes leaving the device in its original state without disabling any features, such as internet or network access. There are ways for attackers, however, to wipe mobile phones from a remote location, requiring a phone to be kept off networks and the internet until a full forensic analysis has been completed. As will be discussed, there are many scenarios where modifications to the device are a necessity for data security. In any situation, careful consideration of any changes to a device must be documented, repeatable, and verifiable if they are going to hold up in a court of law.

The actual collection of physical evidence is the third step in the analysis process. This can involve the collection of extremely valuable data including emails, photos, user accounts, and much more. Mobile phones have developed into miniature computers, allowing smartphone users to add everything from email accounts to personal banking applications on their personal devices. Much of the data left behind by these applications is accessible through the data collection process. The overarching goal of the third step is the retrieval of that data.

The fourth step in computer forensics, data recovery, is the process of discovering hidden or deleted data on a device. One way to accomplish this feat is through data carving. Data carving is the process of looking on a hard drive or other media for previously deleted data that has not been entirely

overwritten. This data is recoverable because of the way most file systems write data to a hard drive. For example, when a file is written to the hard disk in the NTFS file system, a record of that file is created in the master file table (MFT). This process can be compared to a chapter or page number in a book's table of contents. The MFT points to the physical location of where a file is located on the hard drive just like a table of contents points to a particular page in a book. When the file is requested the computer searches the MFT for the exact location of the file, effectively reducing the time required to find the correct file and ultimately reducing the total time required to execute of the data requested. When the file is deleted, the file itself is left on the disk. The only thing deleted is the entry in the MFT. Thus, the data is still on the hard disk. The only way this data is lost is if the computer writes over it with another file. Even then, if the file is not completely overwritten part of the file is still available to be found.

Another form of data that can be discovered is data outside the allocated space on the hard drive. Also known as unallocated data, this data can include former data 'deleted' on a hard disk as well as data hidden by a computer expert trying to circumvent an incomplete forensic analysis. Both of these types of data can be invaluable to a forensic examiner and should be included in an analysis if it is determined that it could be of value.

According to Palmer, the fifth step in a forensic analysis is the actual analysis of the files. This step is crucial as it allows the analyst to "recreate a specific chain of events or user activity… search for key words and dates… compare and contrast computer code to determine whether a particular program is original or copied from a similar program… [and] advise on what evidence is likely to be found on the computer media and identifying the most effective set of data to search." [9] Traditionally this is the most detailed and labor intensive step, as it is the physical act of searching the forensically obtained image of the device to discover the intended data. A forensic examiner should prepare to spend a majority of their time during the forensic process in this step in order to ensure they have viewed all potential evidence available on a device's image.

The final step is for the examiner to recap what was found for any and all interested parties. This is a crucial step for forensic examiners, as they need to be able to account for all steps in their data extraction and recovery process in a sound forensic report. If they cannot provide a report in such a way, their analysis and expert testimony may be deemed inadmissible in a court of law. While standards for proof in this step are much lower in a corporate environment, this is still an extremely important step to be cognizant of, even during each of the other five steps.

While Palmer originally wrote his six steps for computer forensics, the steps nonetheless apply to mobile forensics as well. They will be applied to this study for both the target Android and iPad devices. They will be modified or redefined as needed, however, as this study proves both computer and mobile forensics are very similar. Many of the theories used by forensic examiners for sound computer forensics directly apply to the emerging field of mobile forensics.

Android Forensics

Step 1: Deciphering the OS and its File Structure

The Devices, Components, and Their Intended Uses

According to informationweek.com, there were 300 million Android devices in use as of February 28, 2012. While that number may seem astounding (that is almost the equivalent of an Android device for every person currently living in the United States), it is even more astounding to think that their market share continues to grow. 850,000 Android devices are being activated daily. [10] These

devices are typically either tablet devices or mobile phones on one of a variety of carriers (Sprint, T-Mobile, Verizon, and AT&T, among others).

The Android operating system is an open-sourced operating system (OS) whose growth has thrived due to its customization. Android's developer, Google, allows manufacturers access to the Android source code to customize as they please. In doing so they allow manufacturers the ability to modify the OS to a consumer's needs while still incorporating Google search functionality. Google profits from the Android marketplace as well as advertisements through Google search, while manufacturers get a low-cost OS customized to their target consumer. Andrew Hoog discusses that "by creating a mobile OS that meets the demands of the consumer as well as the needs of the manufacturers and wireless carriers, Google has an excellent distribution platform for their revenue-generating search and advertising business." [6]

The high amount of customization directly impacts Android forensics. Due to the overwhelming amounts of variation in components, devices, and their intended uses, there is a large degree of variation that a mobile examiner has to be prepared for when examining an Android device. For example, most Android devices now have the ability to communicate with GPS, or the Global Positioning System, however some do not. They also may or may not include Wi-Fi, Bluetooth, front and rear cameras that have the ability to take photos and videos, gyroscopes, speakers and other features. [6] The immense amount of features and customization available with the Android OS presents distinct challenges for forensic examiners.

The Operating System and the Dalvik Virtual Machine

The customization described earlier does not end with device components. Manufacturer customizations to the Android operating system have created multiple versions of Android operating systems. To date there are 4 different Android OS versions, the most recent being named Android 4.0 and codenamed 'Ice Cream Sandwich.' Each version has multiple versions of their own customized by manufacturers and everyday users alike. Forensic examiners need to be aware of what version and kernel the device is running. According to Andrew Hoog "an HTC Dream 100 running Android 1.5 with kernel 2.6.30.4 or earlier is vastly different than the same device running Android 1.6 or a kernel greater than 2.6.30.4." [6]

Before obtaining a successful forensic analysis of an Android device, the examiner must have a complete understanding of the unique file structure and data storage elements on an Android device. Android devices utilize two main types of memory: random-access memory, commonly referred to as RAM, and NAND flash memory. RAM is volatile, which means that it does not preserve its state without power, whereas NAND is nonvolatile, which means that it will preserve its state even in the absence of a power source. The NAND flash is where an Android's operating system and user data is stored. [6] Because of this, the NAND flash is used primarily for data storage. NAND flash and its file structure are a treasure trove of data for forensic examiners and will be a focal point in this study.

The Android OS uses a unique application runtime sandbox known as the Dalvik virtual machine. This virtual machine structure attempts to increase security in Android devices by separating individual applications from one another. Each application installed is "assigned a unique Linux user and group ID and runs in its own process and Dalvik VM." [6] The system then uses the individual application's user and group ID's to allow requested data and component permissions. Also, when an application is installed on a device the system gives the application its own directory. Applications are not able to access another application's data unless given explicit permission. That permission is typically only granted by user modification of a device or, in rare scenarios, allowed by a developer. While this means

that a forensic examiner may have to look in multiple different folders for the correct device data, this structure can also be a valuable asset. If the examiner knows the application that used, stored, or executed the data that the examiner is looking for, they may be able to quickly identify the correct data folder they require.

The Software Development Kit (SDK) and its use in Android forensics

Google's dedication to a completely open-source platform has resulted in the creation of the Android Software Development Kit, or SDK. The SDK is an application development resource that can be used to develop Android applications. Included in the SDK are software libraries, application programming interfaces, and a full-fledged Android emulator, among other things. [6] The SDK can be a valuable tool for forensic examiners as it provides an examiner with the ability to manipulate and examine an Android system, including its data structure, without repercussion. The emulator allows a user to load any Android OS that has been created. These versions may include customized OS's, such as those specifically created for tablets or customized versions for specific mobile phones. While this is a valuable tool for many forensic examiners, it did not prove as helpful for this study as originally hoped. In that regard, this study will conclude with an overview of the SDK, how to install it, and why it was not practical in this situation.

Step 2: Data Preservation on an Android Device

As discussed earlier, Palmer recommends that a forensic examiner ensure that potential evidence is not damaged, viruses are not introduced, data is protected from outside damage, and a proper chain of custody is maintained during the process. This is one of the most important, and in many cases most difficult, aspect of mobile forensics. RAM on a mobile phone is volatile, just like with a desktop or laptop PC. If a mobile phone is turned off or loses power, any data stored on the mobile device is lost. To prevent this scenario an examiner may want to plug a device in to allow it to charge, preserving the volatile data if the device is on. To do so, however, would allow the device to be continuously exposed to Wi-Fi and cellular network. Leaving the device exposed to network signals could prove detrimental to a forensic examination if an attacker has the ability to wipe a device remotely.

To prevent against this type of attack the forensic examiner must disable network communications. This could include putting the device in 'airplane' mode or manually shutting off each of the device's radios. Any changes made to the phone must be carefully reviewed, documented, and must be repeatable or they may be challenged in a court of law.

One popular tool used by examiners to avoid changes to a device is known as a Faraday bag. The Faraday bag is used to isolate the device's signal without modifying the device by denying all network signals into or out of the bag. While original versions of the Faraday bag did not allow charging, preventing the analyst from charging the phone, new Faraday bags have charging capabilities. While preparing to use the Faraday bag the examiner needs to ensure that they have the correct charging adapters for any phone that they may come across.

Step 3: The Forensic Examination of an Android Device

The Physical Data Examination

A proper physical analysis of an Android device depends on where the physical data to be examined resides. Traditionally, Android devices store physical data on unencrypted SD cards. An

analysis of these cards can be as simple as removing them from the device, plugging them into a card reader attached to a write-blocker, and using a forensic tool such as AccessData's FTK Imager and Forensic Toolkit to examine their contents. Newer Android devices, however, have 'virtual' SD cards where a portion of their built-in hard drive space is segmented as an SD card. In other instances, where the device has a physical SD card, the card is stored underneath the device's battery, as is the case with the HTC Thunderbolt that will be analyzed in this study. In a situation like this, if it is not practical to remove the battery for fear of loss of volatile data, there is another way to obtain physical data without turning off the phone, including mounting the phone to the host device as a disk drive.

Setting the Phone to 'Mount as a Disk Drive' Mode

To set the phone into its 'mount as disk drive' mode the analyst must set the phone to act like a hard drive mounted to the host system via Universal Serial Bus, or USB, when connected to a computer. While this is not traditionally a default setting on the phone, some users may leave their phone in this setting. If the phone is already set to 'mount as disk drive' mode an examiner would be able to complete a forensic analysis of the physical data from the device without modifying the phone in any way. If the device is not set to automatically mount as a disk drive, however, the forensic examiner must access the phone to change its settings in order to obtain a physical analysis. In this situation the examiner must be extremely careful when making any modifications to the phone to ensure no unnecessary modifications are made that may result in a dismissal of evidence. Furthermore, if the phone is password protected the examiner would be unable to set the phone into the required mode until they successfully bypass the password.

Obtaining a Disk Image

The next step is to take an image of the target hard disk. After successfully setting the phone into 'mount as a disk drive' mode, the forensic examiner will connect the device to whatever computer they are going to use to complete their forensic examination (if they have not done so already) and load the imaging software of their choice. This study will use AccessData's FTK Imager to complete the disk imaging. Figure 1 is a screenshot of the 'mount as disk drive' option in the pull-down bar at the top of the test device.

Figure 1: Connect to PC options on the test Android device

Once the device to be examined is in the correct state for imaging, FTK Imager makes it quite easy to take a full physical image of the device. After connecting the device via USB, the examiner should select the 'Create Disk Image' option under the File menu in FTK Imager (see Figure 2 for an example).

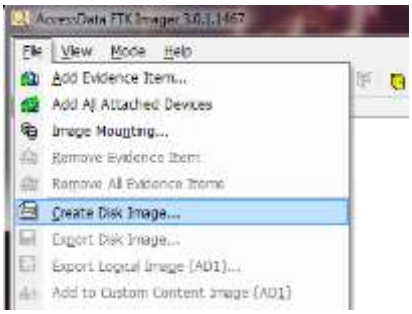

Figure 2: Create a disk image (FTK Imager)

As shown in Figure 3, the next step is to select 'Physical Drive' as the source evidence type.

Figure 3: Select source (FTK Imager)

The device to be imaged should be listed as one of the physical drives available in the 'Source Drive Selection' drop down list. The examiner should then select the targeted drive and click 'Finish' (see Figure 4).

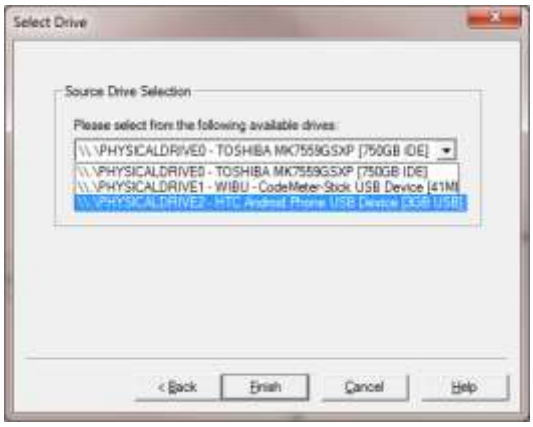

Figure 4: Select drive (FTK Imager)

As shown in Figure 5, the next option to select is the destination image type. Each option has its own unique output types and attributes that make them desirable for a number of various forensic platforms. Since FTK reads each image format, for the purposes of this study selecting any type in this list is acceptable.

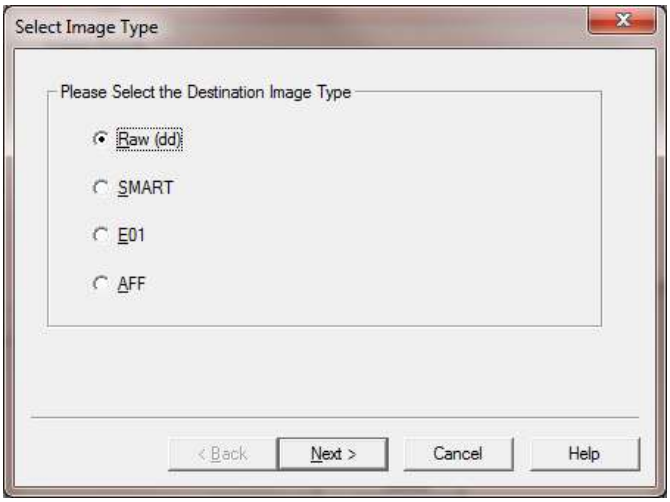

Figure 5: Select image type (FTK Imager)

The final options in FTK Imager correspond to case identifying information such as case numbers, evidence numbers, and examiners (see Figure 6).

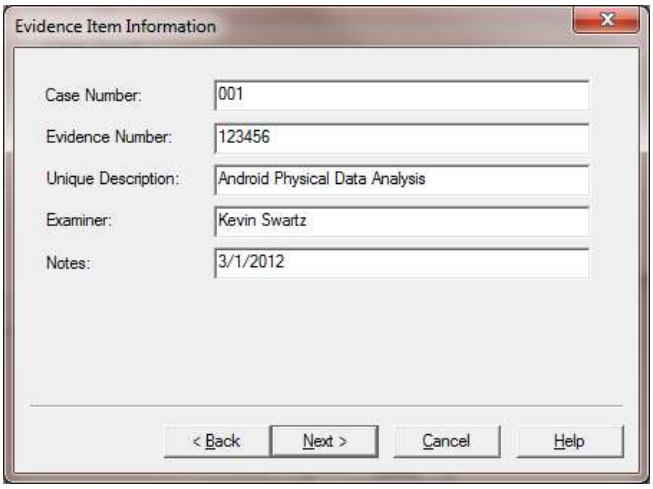

Figure 6: Evidence item information (FTK Imager)

After adding all that information, selecting a target output folder, then selecting 'Finish', FTK Imager will complete a physical analysis of the SD card (see Figure 7).

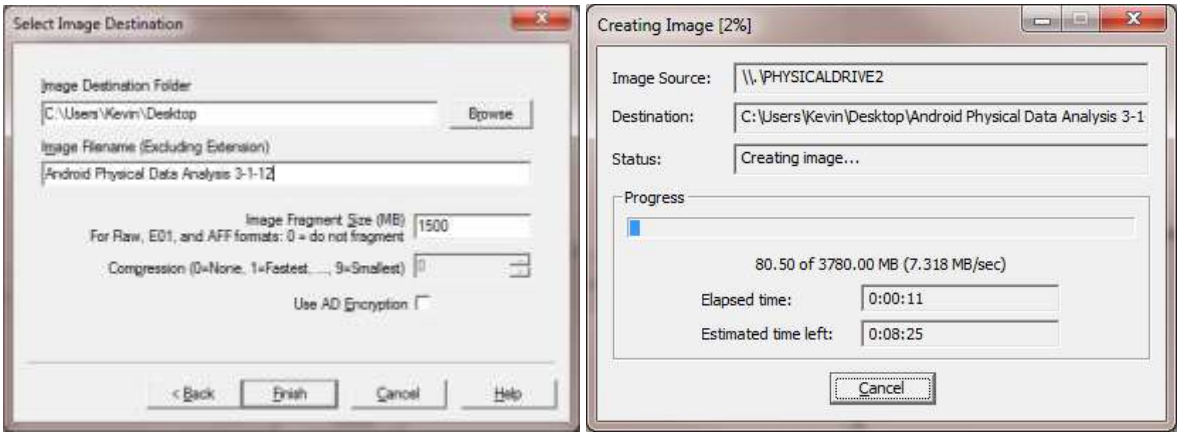

Figure 7: Selecting the image destination (Left), Creating the disk image (Right) (FTK Imager)

The Logical Data Examination

Setting the Phone to USB Debugging Mode

While a physical analysis of an Android device can require slight modification of a device, a logical analysis of an Android device may require an entire modification of the host OS. At the very least a logical analysis will typically require the use of device drivers and OS specific software that will connect to a targeted device and, ultimately, make changes to the device as well.

Just like the 'Mount as Disk Drive' option is vital to a physical analysis, the USB debugging option on the target device is critical to enable a logical analysis of a device. Furthermore, just like the modifications required for physical analysis, enabling USB debugging can be extremely difficult to achieve if the device is locked. Even if USB debugging is accomplished, more modifications may necessary to obtain a full logical analysis of an Android device.

To enable USB debugging on a traditional Android OS, first select 'Settings' on the device's home screen. Once in the settings menu select 'Applications', then 'Development'. From there select the 'USB debugging' option. This will bring up a very straight-forward warning about the dangers of allowing USB debugging, including reading log data and copying data between the Android device and a separate computer (two things that are of great importance to a forensic examiner). Figure: 8 contains detailed screenshots of this process on an Android device.

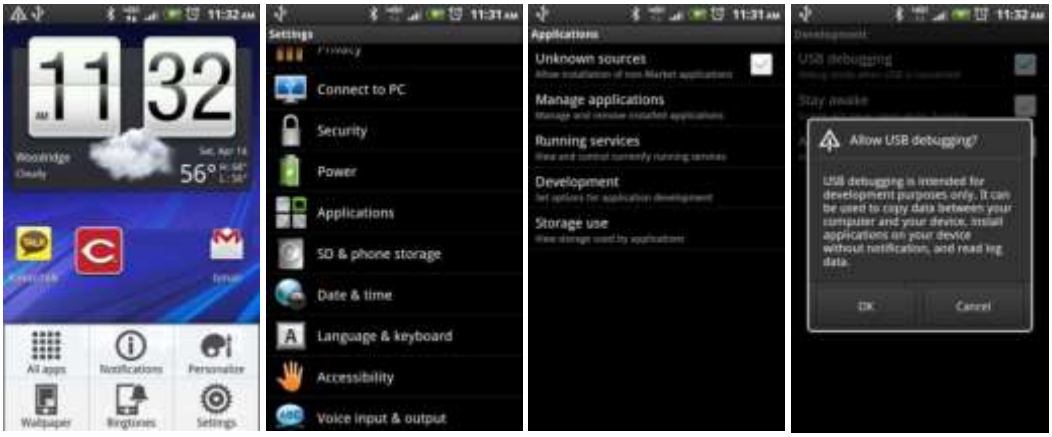

Figure 8: Enabling USB debugging on an Android device

As seen in Figure 8, the symbol in the top toolbar on an Android device that looks like an exclamation point partially inside a triangle indicates that the device is already in USB debugging mode. A symbol that is visible on the lock screen (as is shown in Figure 9) indicates that the phone is already in USB debugging mode. This also indicates to the examiner that they will not have to unlock the device to enable 'USB debugging mode'.

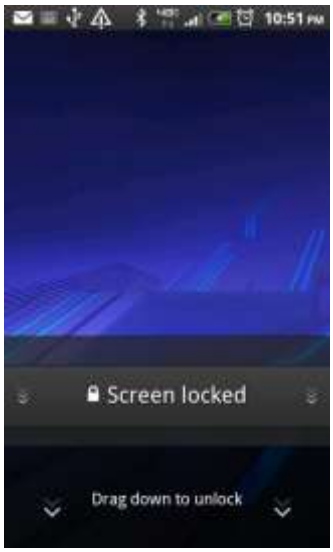

Figure 9: USB debugging symbol shown on lock screen (top bar, fourth icon in from the left) (Android)

Specific drivers and other software may also be required on the host system that is being used to complete the forensic imaging. One such program, known as the Android Debug Bridge (ADB), is required for most logical examinations of Android devices. The ADB is used by the host system to communicate with an Android device. It is included in the Software Development Kit discussed earlier and can be run in Windows or Linux with the 'adb devices' command (as seen in Figure 10). If the device to be examined is in USB debugging mode and correctly connected to the host system, once the adb debug command is run the target device will show up as an available device by its unique serial number (the serial number in Figure 10 is covered for security).

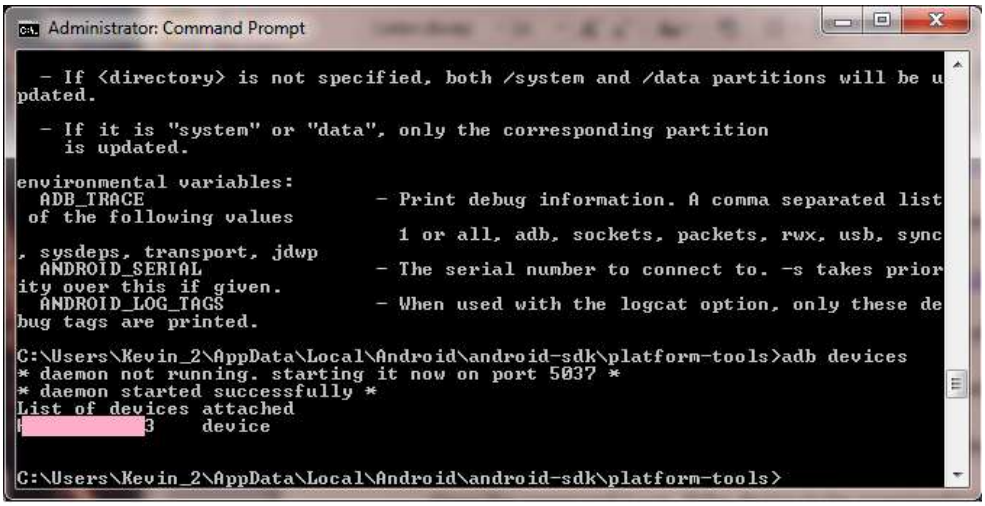

Figure 10: The ADB as run in Windows

To Root or Not to Root?

While it would be ideal that these two options alone make a logical analysis of a device possible, this is typically not the case with an Android device. The unique file structure in the Android OS prevents applications from reading, modifying, and manipulating data of other applications through the use of the Dalvik VM. This file structure can also prevent the examiner of a device from viewing valuable data. Because of this, the forensic examiner may have to gain root access to the device to obtain targeted data. [6] Root access is the rough equivalent to administrator access on a Windows device. It allows an owner full control to read, modify, and write data to their device instead of the restricted control that they have out of the box.

As rooting techniques have become more common, the process of rooting a device has become much simpler than it has been in the past. For example, to root the HTC Thunderbolt for this study a step-by-step guide and all required files were available in one online location. [11] After thirty minutes (mostly consisting of waiting for the device to update and upgrade itself) the device was rooted and a full logical analysis was possible.

Rooting a device is the most effective first step to obtaining a full logical analysis of a device. Even so, there are many dangers that forensic examiners must be aware of. As discussed earlier, changes to a device during a forensic examination should only be made by a trained professional if absolutely necessary. Furthermore, those changes should be repeatable should another equally qualified examiner want to recreate what changes took place for verification. Serious legal questions could arise with the forensic analysis and all credibility gained in the evidence could be lost (including having the evidence removed from a case) if those guidelines are not followed.

Though rooting a phone may be permissible in court if it is proven that the modification to the device was necessary, it may not always be possible. The only way to root the HTC Thunderbolt that was used in this research was to downgrade the OS from its most recent firmware update of 2.11.605.9 to a previous update of version 2.11.605.5. Doing this downgrade wiped the device clean, erasing all data that would have been crucial to the forensic analysis. Therefore, while data was gathered from the device after social media accounts were loaded back on, this analysis technique proved to have serious consequences if not used correctly.

If it is not possible to root a device, more invasive measures are available that can be used to obtain a logical analysis of data. Typically these examinations are much less efficient, as they may require a full disassembly of the target device to directly access the hard drive. This would require equipment specifically created to read individual NAND flash chips. As such, these measures are outside the scope of this study.

The Logical Examination

After successfully gaining root access to the target device, putting it into USB Debug mode, starting the ADB, connecting the target device to the host forensic machine while using a write-blocker, and ensuring that the device has been read correctly by the host forensic system's Android Debug Bridge, the device is ready for a logical analysis. There are many open sourced and commercial applications that are able to complete a logical analysis of an Android device. This study will use a commercial tool from AccessData known as the Mobile Phone Examiner Plus (MPE+) as the tool of choice.

The MPE+ is a versatile mobile forensics application that can be used to obtain data from over 3,500 phones and more than 80% of all CDMA handsets. [12] It is able to obtain logical data of rooted Android devices as well as other important data such as SMS and MMS messages, contact lists, call history, carved data, SIM and USIM card data, and more.

As seen in Figure 11, after loading MPE+ and selecting the proper targeted device (in this case, the 'Mobile Device' option), MPE+ loads a Device Selection screen that allows the examiner to select the specific manufacturer and model device that they are examining, which is shown in Figure 12.

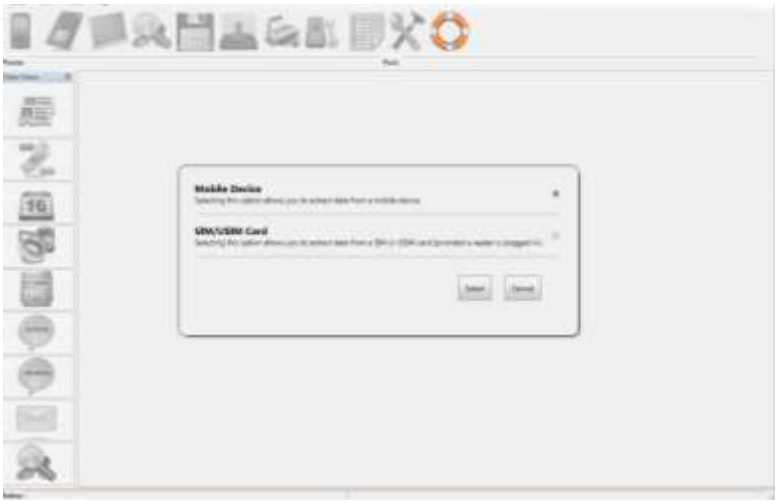

Figure 11: The data source extraction window (MPE+)

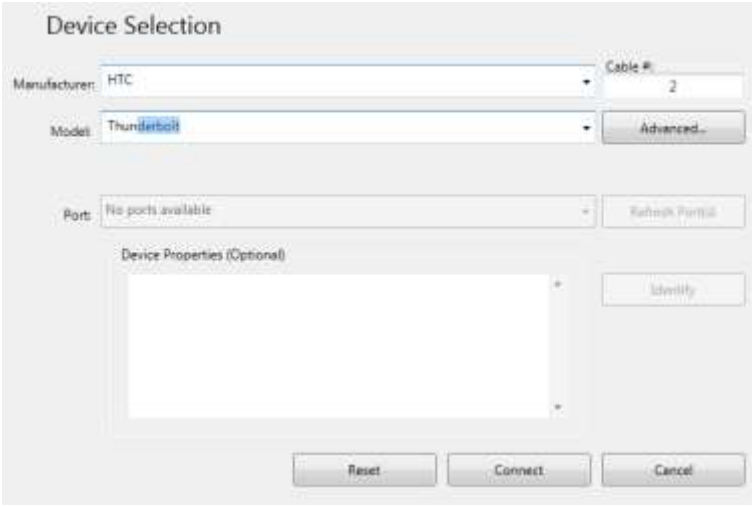

Figure 12: The Device Selection screen (MPE+)

Since the analyst has already prepared the device to be analyzed, simply selecting the specific phone manufacturer and device model options under the aptly named manufacturer and model settings will put the MPE+ in the correct state for examination (for this study's HTC Thunderbolt test device the selections were 'HTC' and 'Thunderbolt' as shown in Figure 12). After highlighting the specific logical data to target and selecting the 'Extract' button, MPE+ will connect to the device and begin the logical analysis.

Step 4: Data Recovery

As discussed, one highly effective method of data recovery is through data carving. Data carving is a data processing option in FTK which allows the user to search for files that are incomplete or are no longer part of the traditional file structure. This can include BMP, GIF, JPEG, and HTML file types, among others. [13] For example, a photo that has been deleted and partially over-written may be found during the data carving process. In this example, FTK will look for the portions of the photo that have not been overwritten and will catalogue them as individual evidence files in the case. [14]

This can be an incredibly useful tool for a skilled mobile forensic analyst. Since many phones now double as both cameras and GPS systems, many photos and files are tagged with GPS coordinates. Deleted photos that are carved may contain GPS coordinates in their EXIF data, providing an examiner with compelling evidence that an original user may have thought was deleted from the phone. This can also be a danger for individuals, as will be discussed later, because sensitive location data may be available in photos that users thought were gone but are recoverable by any attacker with the correct software (such as FTK). By selecting the "Data carve" option during case pre-processing in FTK the forensic examiner has told FTK to search for incomplete or deleted files as well as files that are in 'slack space'. Figure 13 shows this option as it appears in FTK while Figure 14 shows a side by side of a two carved files – one almost completely unreadable, the other perfectly intact. Both carved photos are of the same JPEG file stored in two different locations.

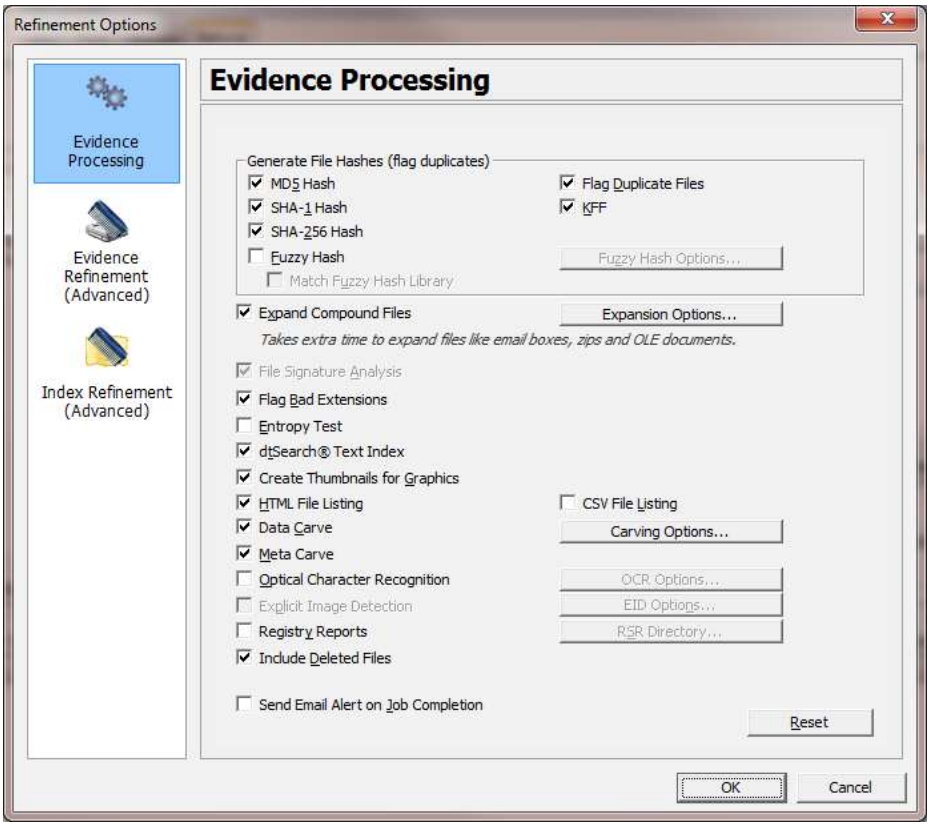

Figure 13: Evidence processing options (FTK)

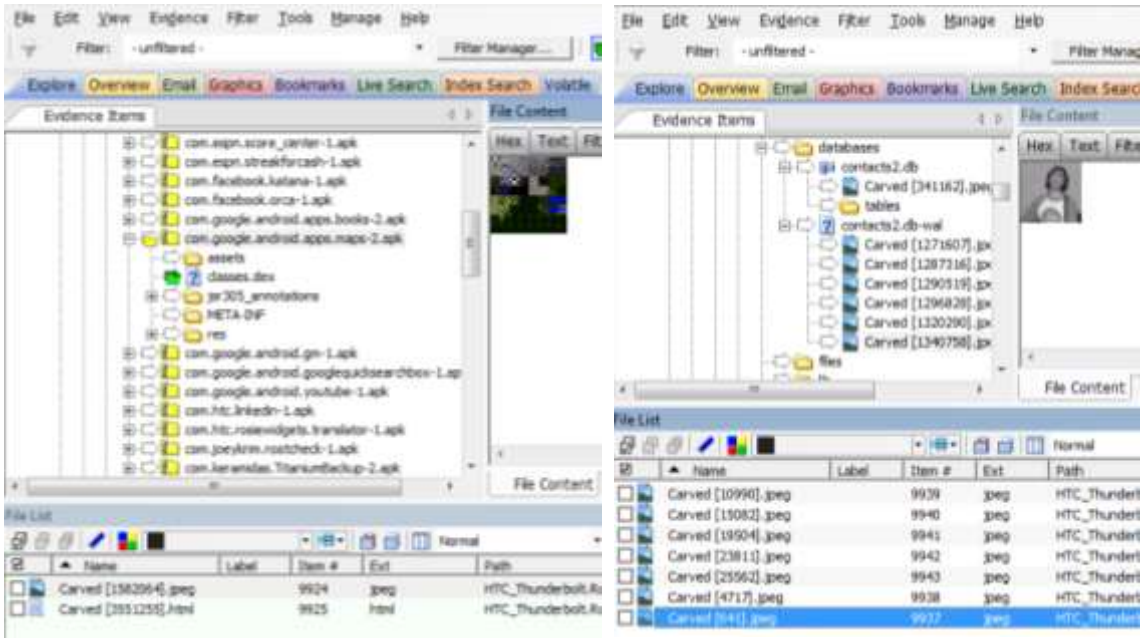

Figure 14: Files carved using the data carve processing option (FTK)

Step 5: Analyzing the Results

Analyzing the Data from the Physical Image Using FTK Imager

The FTK Imager tool used to create the forensic physical image of the Android device can also be used to preview the data discovered on the device. This is especially important when trying to decide if changes to the device (shutting it off, for example) are acceptable. For example, if the examiner knows the data they are looking for is in a specific folder, FTK imager will allow them to see the folder in question. If the folder is accessible and the data that they are searching for has been forensically retrieved the examiner can immediately import the image into the Forensic Toolkit (FTK) for further analysis. This can save critical time if a battery is running low on a device that cannot be charged.

Analyzing the Data from the Physical Image Using FTK

The physical analysis of the Android device's imaged SD card provided a great deal of confidential information. Since mobile devices are multifaceted tools and may be used to take photos, videos, audio recordings, and provide GPS coordinates (among other things), a forensic examiner should ensure that they obtain all possible data from a device.

This section is going to detail the results of the physical analysis of the Android device, also known as the HTC Thunderbolt (or simply HTC or Android from here on out). It will include file locations that contain valuable data and will provide examples of what an examiner (or attacker) can expect to gain from those folders.

The Evidence Tree

One of the most important aspects for a forensic examiner to understand about a device is its file structure. There are traditionally two main types of evidence folders an examiner will see after a physical imaging of an Android device (there may be cases of devices that have multiple partitions, however that scenario is uncommon. In that instance, each partition will be individually numbered and

listed and the same forensic analysis rules that will be discussed apply to both partitions). The first will be the hard disk partition labeled as 'Partition 1'. The second is labeled as 'Unpartitioned Space'. Both of these evidence types are shown in Figure 15.

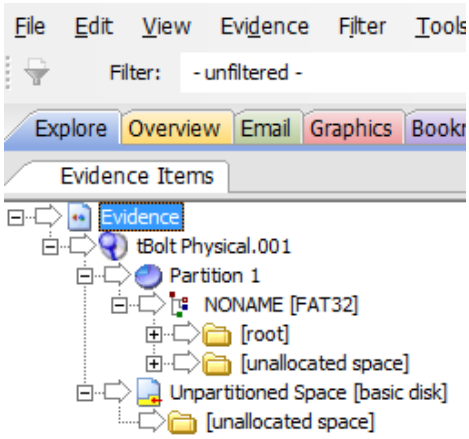

Figure 15: The file system from the physical analysis of the test Android device (FTK)

There are also two main sections on the disk partition that are 'carved' during a forensic analysis. The first is the disk itself, just how it would appear to a computer system if it were connected and read. This data resides under the [root] folder (from here it will be described as the /root folder). The second section contains data carved from unallocated space on the hard disk. This may include data that was deleted from the hard drive but never over-written and recovered during the forensic analysis.

The /root folder can be a 'treasure trove' for forensic analysts and attackers alike. Under the /root folder is the /.bookmark_thumb1 folder. This folder contains jpg images of websites visited by the Android device. In this study, Google searches, Lewis University logon webpages, and Facebook profile pages were among the photos stored in this location. These are photos of websites that were bookmarked on the target Android device. When a user bookmarks a website, the phone will regularly update the thumbnail photo of the website to one of the last views that a user had of it. This can contain valuable data, as described above. See Figure 16 for an example.

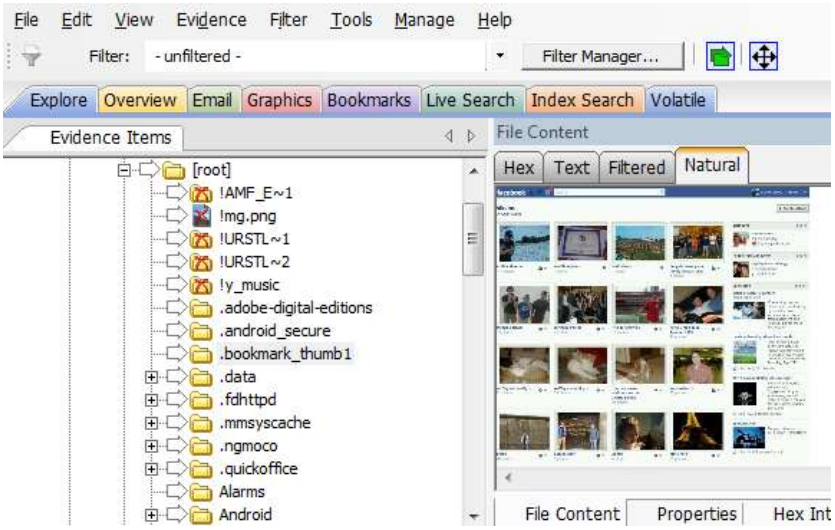

Figure 16: A thumbnail from the test device displaying the user's Facebook photo albums (FTK)

The /root/android/data/com.google.android.apps.maps folder contains user data relating to the Google maps function on the Android device. On this study's test device, there were two audio clips left on the device that replayed turn by turn directions originally requested by the user. This also contained the /cache_r.0 folder which housed the images of every street the Google maps device has recommended to this device's user.

Traditionally, the folder that holds photographs taken by the device is titled "DCIM", and the study's Android device is no different. Inside the /root/DCIM folder are two separate folders, as shown in Figure 17. The first contains thumbnails for every photo and video stored on the device and is properly named '.thumbnails'. The other folder, named '100MEDIA', contains photos and videos taken on the device or added later by the user.

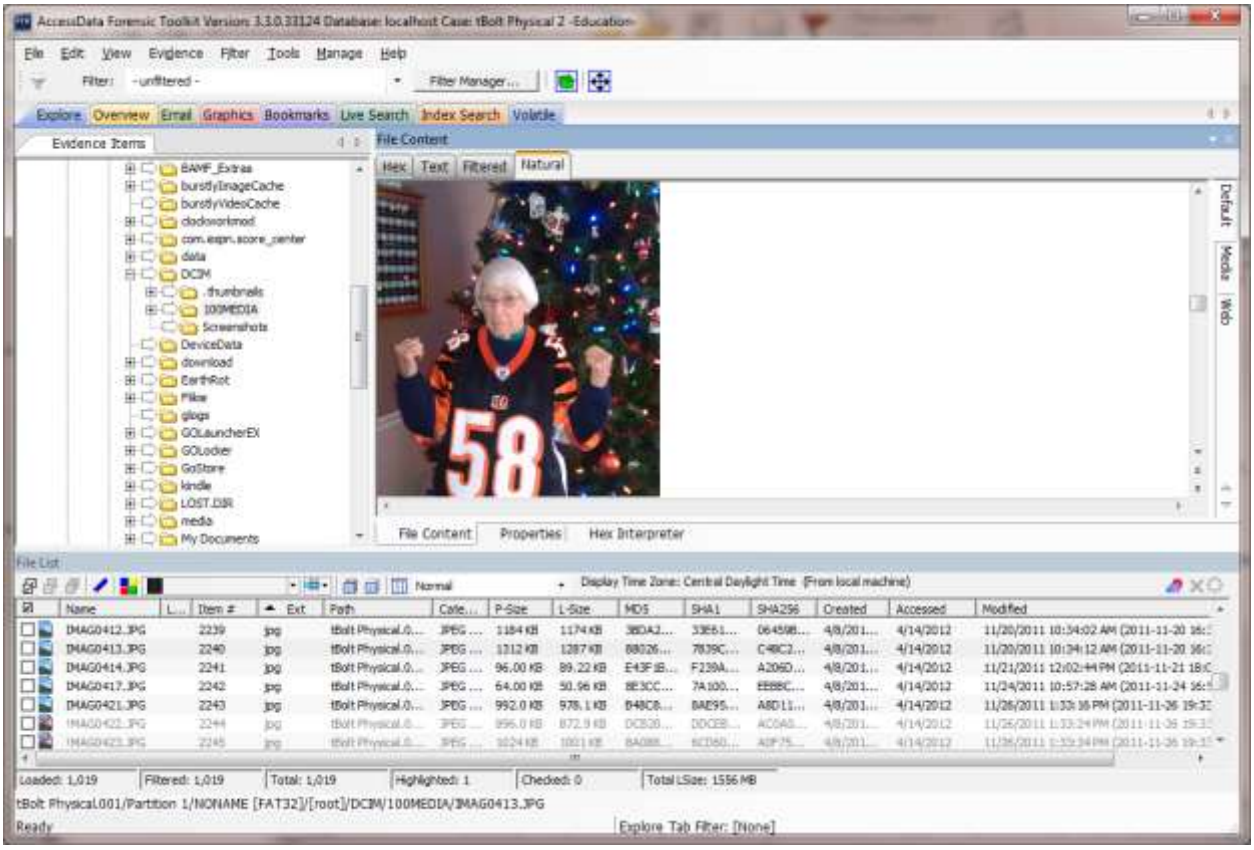

Figure 17: A file from the DCIM folder (FTK)

The photos stored in the 100MEDIA folder can contain extremely valuable data for an analyst. Many phones are capable of utilizing the Global Positioning System, also known as GPS. As discussed, this feature can also be used to tag photos with GPS location data. This information is stored in the 'EXIF' data attached to each photo and can be accessed either through a secondary EXIF reader program or in the 'Properties' tab in FTK.

For example, on the test device there is a file named "IMAG0320.jpg". The properties data shows that its latitude and longitude are 42/1 22/1 2548/100 N by 87/1 56/1 2898/100 W. Using the free calculator at http://transition.fcc.gov/mb/audio/bickel/DDDMMSS-decimal.html [15] and plugging the output decimal coordinates into Google maps, the examiner discovered that this photo was taken at Six Flags Great America (see Figure 18 and Figure 19). Other valuable EXIF data includes date and time

stamps, file hashes, and GPS date stamps. For example, the EXIF data shows that the photograph taken at Six Flags was taken on October 1, 2011 at 1:43:24 PM. Since EXIF data also provides the name and device information of the original source (camera, scanner, etc.), it can also prove if the photo was taken on another device and copied or moved to its current location

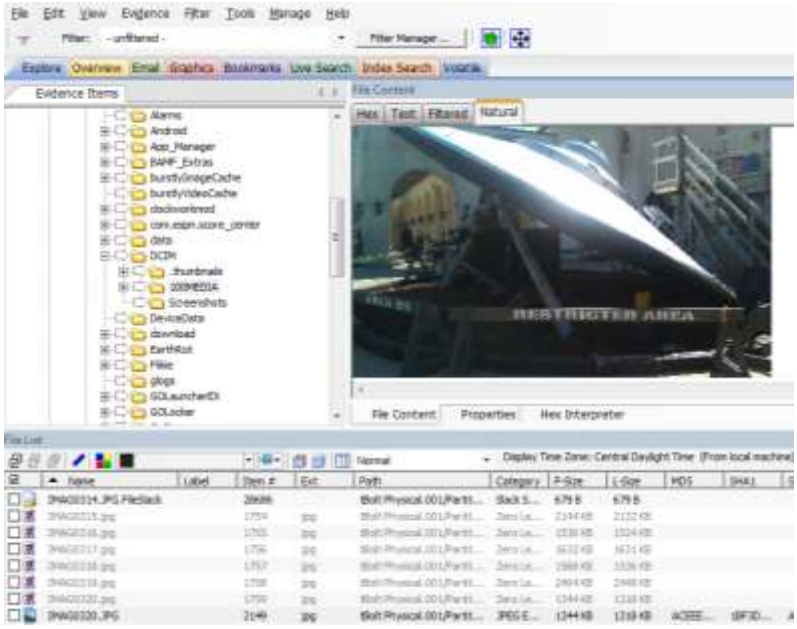

Figure 18: The original photo discovered on the device (FTK)

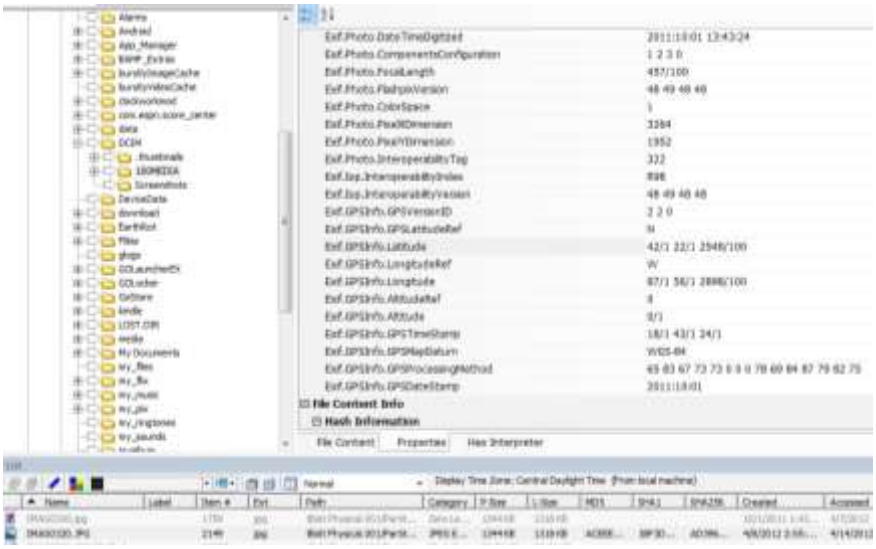

Figure 19: The EXIF data and properties of the photo displayed in Figure 20 (FTK)

Also available in the /100MEDIA folder are videos that were either taken on the device or loaded by the user into the folder at a later date. FTK can play most video formats allowing the examiner to preview the videos natively in the application. If FTK is unable to play a video the file can be exported to be played in a video player outside of FTK on the host system.

The /root/download folder contains most files downloaded to the device. This folder is the default location for .pdf files to be downloaded and can be of considerable value to an attacker looking for confidential data. The HTC device contained .pdf's that had long been closed. This is because the files remained were never deleted, therefore they remained on the device. Figure 20 contains an example of a .pdf file that was pulled in its entirety from the /root/download folder.

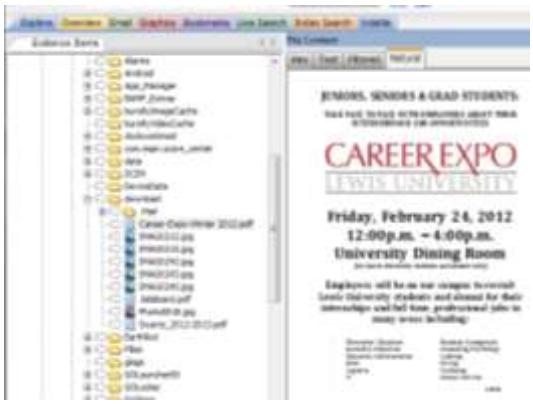

Figure 20: A .pdf file pulled from the target device (FTK)

The /root/MyDocuments/MyRecordings folder is the default location for voice recordings on Android devices. The test device contained multiple voice recordings that were stored in the .amr format. The .amr format is read by QuickTime. In FTK files can be opened in QuickTime by right clicking on audio log and selecting "Open with -> QuickTime".

The physical analysis can also discover data left on the SD card from previous devices. The /root/my_flix, /root/my_music/, and /root/my_pix are not traditional folders on an Android device, however were a source of data of each file type found on the HTC's SD card. It can be inferred that this data was stored on the SD card when it was inserted in a previous phone and eventually was transferred with the SD card when the card was installed into the HTC device. This can be confirmed by examining the EXIF "image.make" and "image.model" data in the photographs (see Figure 21 for an example). These indicate that this was indeed the case for the photos, music, and videos discovered in these folders.

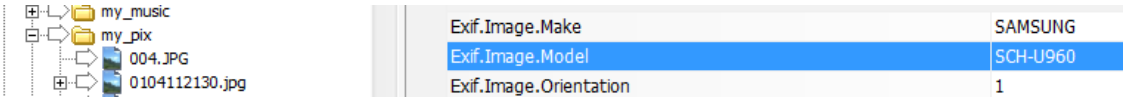

Figure 21: Properties that detail the make and model of the device a photo was taken on (FTK)

The other important locations to note are the unallocated and unpartitioned space. According to Peter White, "The contents of unallocated space (if anything at all) are the remnants of data that must once have existed on the system as 'live' files, and have subsequently been deleted." [16] Data in this location can include anything deleted from the device. On the HTC there were 241 files in unallocated space. All of those files were deleted photos that were recovered in the forensic examination.

It is rare to find data in unpartitioned space, as this is space that is not recognized as a part of the file system and, as such, no data will be written to it by a traditional file system. While advanced hackers can place data in unpartitioned space to try to hide it, it is sufficient for this study to understand

its concept. FTK is able to search unpartitioned space for any remaining data and will carve it as evidence if found.

Results of the Logical Analysis

Analyzing the Data from the Logical Image Using MPE+

Just like FTK Imager for physical acquisitions, MPE+ is a great evidence preview tool for logical Android data. In particular, Access Data imports calendar, SMS, MMS, contacts, and call history data, then presents it to the examiner in an easy to read table format. It also presents the logical data acquired in a folder tree view for the examiner to preview and, if desired, analyze. As mentioned however, MPE+ does not have as many case processing options as FTK and should only be used for case previewing or critical data identification before a full analysis is commenced.

One particular feature to note of MPE+ is the contact list. It lists all contacts found on the device along with their email addresses, the dates and times of last contact, the number of times contacted from the device, as well as any device notes about the contact (see Figure 22).

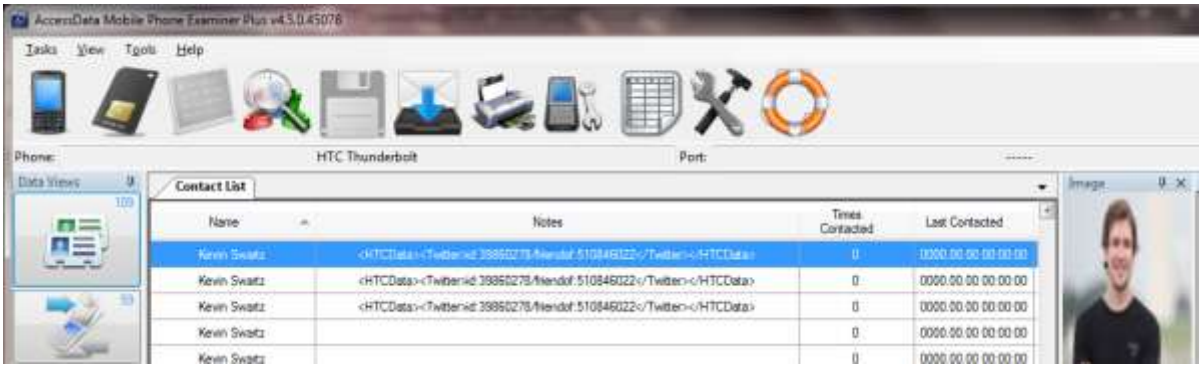

Figure 22: The contacts list (MPE+)

On the HTC these notes included the contact and user's UID numbers. For example, one of the Facebook accounts connected to the device has the user ID of 22900074. The HTC has been granted permission by the user to search their connected Facebook account for any matches between the user's Facebook friends list and their contact list. Any connections approved ('linked') by the user are noted by User ID in the notes section of the MPE+ contact list, and are listed as "<HTCData>John Doe– Facebook ID 22901055 / friendof:22900074<Facebook></HTCData>". It can be reasonably assumed that this data is being pulled from the HTC Data table stored on the device.

MPE+ also details all SMS and MMS messages pulled from the device and presents them in a table format. The columns in the SMS and MMS databases are: Direction (sent or incoming), Date, To (by phone number), From (by phone number), the content of the message, and if the message has been read or not. This can be valuable information, both to a criminal forensic examiner and an attacker, as extremely confidential data can be sent via text message and retrieved via MPE+ in the clear.

Analyzing the Data from the Logical Image Using FTK

The Role of Databases in Mobile Forensics

An increasingly popular form of NAND and SD based storage is the SQLite database file type. According to Andrew Hoog, "SQLite is popular for many reasons. Notably the entire code base is of high

quality, open source, and released to the public domain. The file format and the program itself are very compact and pack significant functionality in less than a few hundred kilobytes." He continues on to state that "the SQLite files are generally stored on the internal storage under /data/data/<packageName>/databases." [6] This is the type of file format that MPE+ pulled for the contact list notes data. The /data/data/ folder is also where a significant amount of the personal data that this study is attempting to discover will most likely be found and, as such, it will be referenced frequently. It is also a folder that is unable to be accessed through traditional forensic methods on an unrooted Android device and is traditionally only viewed via a successful logical analysis of an Android device.

Discovering Artifacts by Application:

Facebook

As discussed, SQLite databases are becoming more common as storage locations for applications on mobile devices. The /data/data/com.facebook.katana/databases folder contains two such tables – fb.db-wal and webview.db – that contain extensive user data. Included in these tables is the device owner's full name, web address links to the owner's Facebook friends' profile photos, Facebook messages in clear text, time stamps for messages and user Facebook actions, as well as the device owner's profile information (including their Facebook profile's listed education, workplace, current city, etc.). The tables also contain links to the device owner's Facebook account as well as their connected friend's Facebook accounts via user ID (UID). Similar data can be retrieved in the /data/data/com.android.browser/app_databases/localstorage folder under the http_m.facebook.com_0_localstorage database. Figure 23 shows user data found in clear text from the fb.db-wal file. User data discovered in clear text, including the user's first name, last name, UID, and Facebook login email address is highlighted. Session keys can be found in the Figure as well.

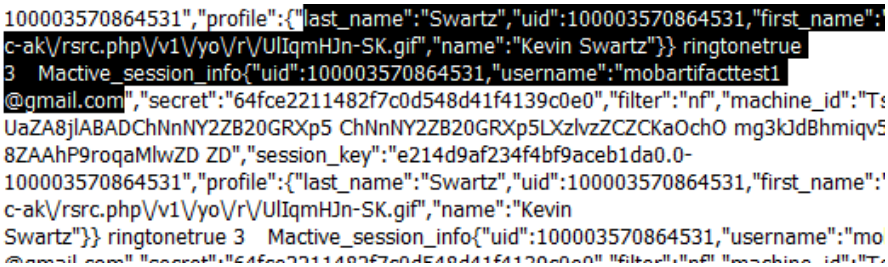

Figure 23: Confidential user data in clear text from the fb.db-wal file (FTK)

The /data/data/com.facebook.orca/databases folder contains multiple databases with confidential Facebook data as well. For example, the /prefs_db database contains the string "first_name":"Kevin","last_name":"Kevin","name":"Kevin Swartz","emails":["mobartifacttest1@Gmail.com"],"pic_big":http://profile.ak.fbcdn.net/static-ak/rsrc.php/v1/yo/r/UlIqmHJn-SK.gif (See Figure 24). This is the device owner's first and last name, the email account that is tied to this particular Facebook account, and a link to the user's profile photo. The /threads_db database contains Facebook messages sent from one user to another. It also details when the messages were sent with time stamps as well as what users were sending them. It identifies the participants in the conversation by Facebook email, first and last name in clear text, and Facebook UID. The /users_db database contains the email addresses of the device owner's contacts from their contact list. It also connects contacts' email addresses to their unique address book number if possible.

'/orca/ui_counters/dismissed_new_message_nux/timestamp
1331087284993
q/auth/fb_me_user
{"uid":"100003570864531","first_name":"Kevin","last_name":"Kevin","name":"Kevin
Swartz","emails":["mobartifacttest1@gmail.com"],"pic_big":"http://profile.ak.fbcdn.net/static-ak/rsrc.php/v1/yo/r/UlIqmHJn-
SK.gif","pic_square":"http://profile.ak.fbcdn.net/static-ak/rsrc.php/v1/yo/r/UlIqmHJn-
SK.gif","pic_crop":{"uri":"http://static.ak.fbcdn.net/rsrc.php/v1/yL/r/HsTZSDw4avx.gif","width":200,"height":126,"left":0.1850(
":0.8149999
976158142,"top":0.0,"bottom":1.0},"is_pushable":true}

Figure 24: User data from the com.facebook.orca/databases/prefs.db file (FTK)

The data/data/com.facebook.orca/shared_prefs folder contains stored preferences for the HTC device's Facebook application. Parsing through the Hex content in the /com.facebook.orca_preferences.xml file located in the /shared_prefs folder provides the examiner with the most recent Facebook token used (listed under <string name="/auth/fb_token">AAADo1TDZCuu8BANQYhBZAPoE1HtWvLeIgoFtbX2x4ZCYxZBh3jZCAxGSlX45gJzIM</string>), as well as when the authorization expires, the last register and change times for the application, the email address and User ID of the account linked to these preferences, as well as other general preferences used by the application.

The Facebook user data that was found in the notes section of the contacts table in MPE+ may have come from the Contacts2.db file. Found in the /data/data/com.android.providers.contacts folder, this file shows data in the same format that as the contacts columns in MPE+. If the examiner wants to reference data found in the notes column from MPE+ they simply need to reference this file.

The largest collection of Facebook data is located in the /data/data/com.htc.socialnetwork.facebook folder. Two separate files – facebook.db and facebook.db-wal – contain full first and last names of contacts from the owner's Facebook friends lists along with corresponding UID's and links to each user's profile photo. They also contain Facebook and user created groups in clear text. These groups, meant to control what a data a user sees when they log in to Facebook, can provide significantly valuable data to an attacker. For example, if a user only wants to see what's happening with their friends from college (and not their friends from high school or work), they can create a group called "College" and assign their college friends that group title. Facebook automatically creates groups like this for users. Furthermore, Facebook does not encrypt or hide the data while it's sitting in the facebook.db-wal file, leading to group listings available in clear-text that provide an attacker the user's educational history, hometown, current city and occupation, among others. The facebook.db file contains full conversations with UID's, first and last names, and time stamps, all in clear text. The facebook.db-wal file contains wall posts by the user in clear text, Facebook webpages visited by the user, and wall posts made by the user. For example, data retrieved from this file on the HTC included data from when the user 'checked-in' to the Hollywood Blvd Theater in Woodridge. See Figure 25 for an example of the data available in the facebook.db-wal file.

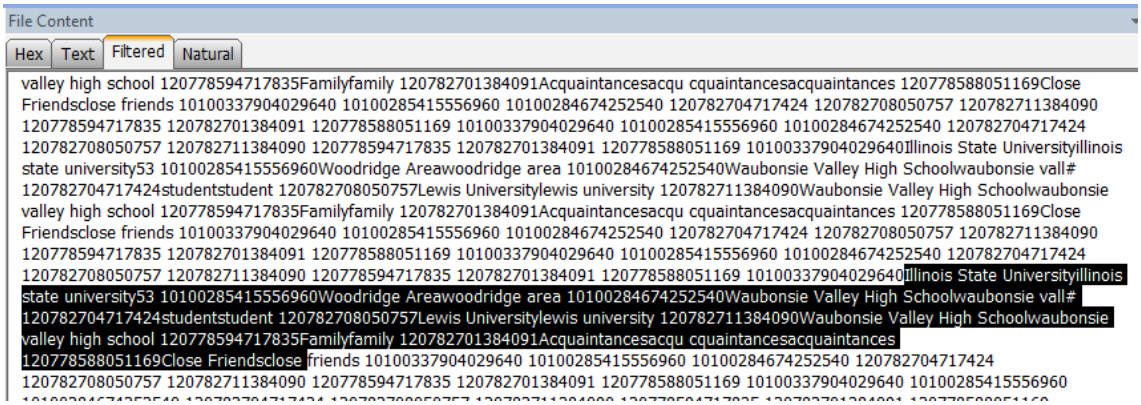

Figure 25: A user's contact groups listed in the /facebook.db-wal file (FTK)

Twitter

Twitter applications use similar data storage methods as Facebook applications. For example, the /data/data/com.twitter.android folder contains a database named "51084622.db". The number "51084622" is the actual Twitter user ID number for the account connected to the HTC device. This database contains tweets made or viewed by the user as well as tweets made by someone that the user is following. Most of these tweets are viewable in their entirety in clear text. For example, in Figure 26 the line "Let's say 199" is an actual post from Twitter user '@mobapptest2' to Twitter user '@redsfan1211' and was retrieved from the 51084622.db file.

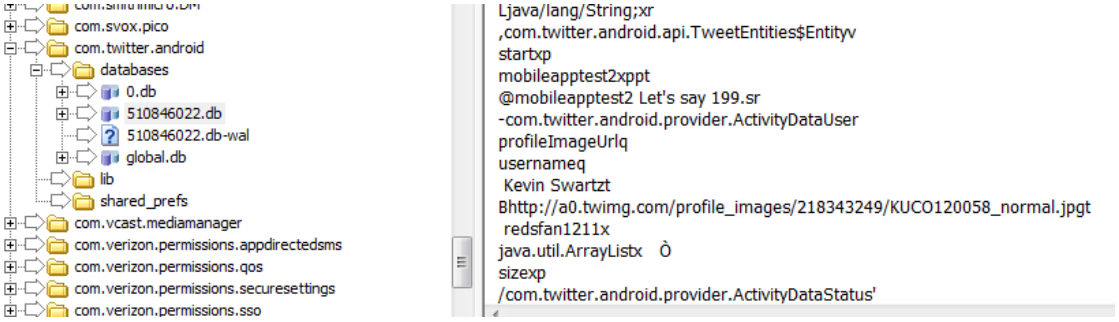

Figure 26: Twitter data found in the 51084622.db file (FTK)

In one of the images taken of the Android device, the /data/data/com.android.browser/databases/webview.db file contained the full username and password of the Twitter account used on the HTC device in clear-text. This was not the case in any of the other images taken of the test phone when it was in a different state than when originally imaged. The viewable password in this scenario may have been a result of a recent web-based Twitter login that was cached in the browser, resulting in the examiner's ability to discover the data during one forensic examination and not others. Once the phone was turned off, the cached data may have been deleted. This discovery is a perfect example of why an examiner should do everything possible to preserve the original state of a mobile device. The string found in the database that correlates to this discovery is: 5+#httpsapi.twitter.com@mobileapptest2testing2222. The '@mobileapptest2' is the username and 'testing2222' is the password tied to the account (see Figure 27).

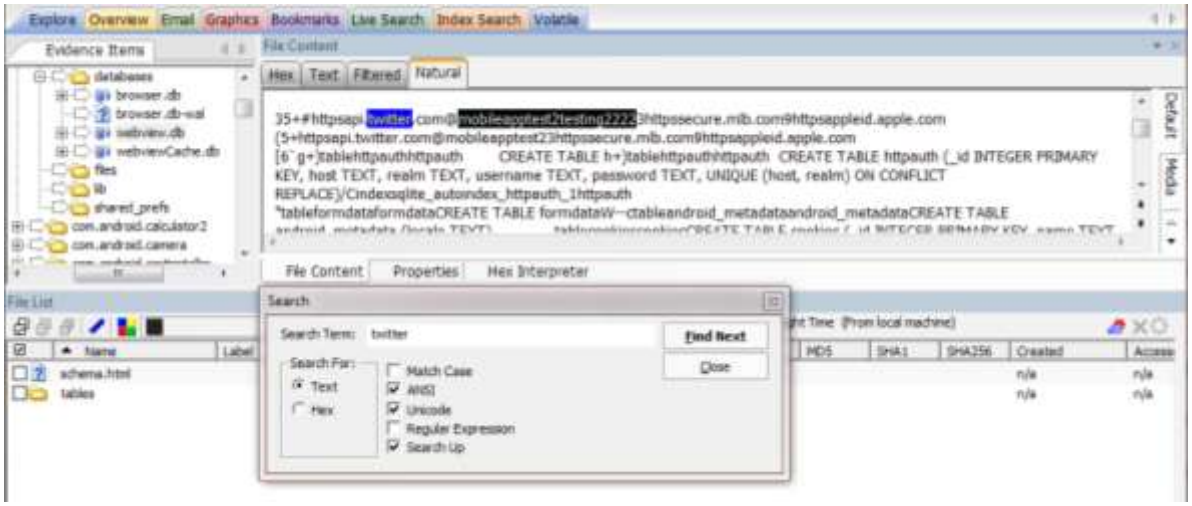

Figure 27: The Twitter username and password recovered from the /webview.db file (FTK)

Just like the Facebook account, the contacts2.db file contains detailed information about a user's Twitter friends, their user ID numbers, and links to their profile pictures. This allows the device to call that data to the phone's individual contact list and supply the contact list with up to date thumbnails for each contact. It also details the user relationships, such as the string "<HTCData><Twitter>id:39460200/friendof:510006022</Twitter></HTCData>", indicating that the individual with the Twitter ID # of 39460200 is a Twitter friend of the Twitter ID # 510006022. This, as will be discussed later, can be valuable data when found in the wrong hands.

The HTC application uniquely tied to Twitter calls the files htcchirp.db and htcchirp.db-wal from the folder /data/data/com.htc.htctwitter/databases/. These databases contain clear-text Twitter data including full conversations, posts, links to user .jpg photos This even includes all posts found on the HTC user's Twitter feed going back to when it was created – a full 6 years before the forensic examination. It also includes many first and last names, Twitter account handles, and Twitter user tag lines. See Figure 28 below for an example of text available in the clear.

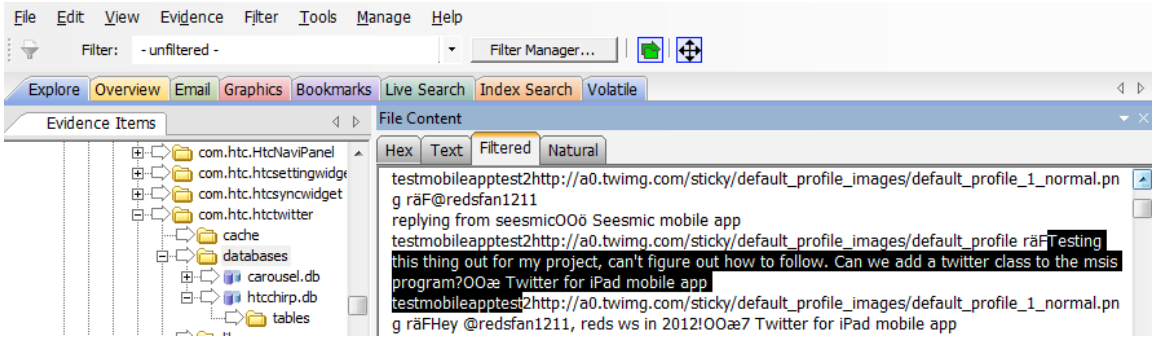

Figure 28: Clear text Twitter posts, names, and links discovered in the htcchirp.db-wal file (FTK)

Seesmic

Seesmic is a third party application that enables users to manage and update multiple social media applications directly from its user interface. For this study, both Twitter and Facebook were loaded onto the Seesmic account on the HTC device. The main Seesmic database was discovered at /data/data/com.seesmic/databases. The individual database that contained the most information was

the Twitter.db-wal file. It contained (contrary to its name) both Twitter and Facebook data. As seen in Figure 29, the file's data included posts made from Seesmic to each of the connected social media websites included 'check-ins' by the Facebook user at a specific location. For example, it contained the Facebook post "Kevin Swartz checked in at the Hollywood Blvd Cinemas" with a link to the Hollywood Blvd Cinema's Facebook page in the post as well. The database also contained personal information from accounts on Twitter and Facebook such as the first and last names of account owners, linked accounts, friends, Twitter names, user ID's for Twitter and Facebook, as well as links to the user's accounts and the photos posted by the user's friends.

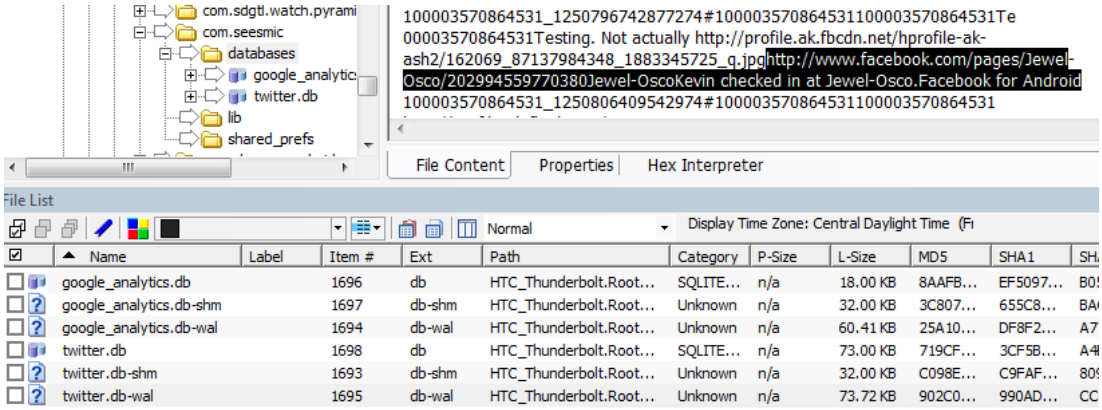

Figure 29: The original user's Facebook post data as well as links available in the twitter.db-wal file (FTK)

Flickr

Flickr is an increasingly popular photo sharing website. The Yahoo!™ based site allows users to upload photos and then share them on multiple different social media websites. This allows a user that would like to post the same photo to both their Twitter and their Facebook accounts to be able to upload the photo once (to Flickr), afterwards sharing it with whichever sites they like. This application has the ability to drastically reduce the downtime required to post photos online, leading to its rising popularity.

Like the previous applications, a great place to start looking for pertinent data in this application is in the database files stored in its individual /data/data/ folder. Flickr's databases are stored at /data/data/com.htc.socialnetwork/flickr. There are two locations and files for an examiner to focus on in this folder: /cache and /databases/webview.db.

The /cache folder contains caches of all .png photos uploaded to flickr. The /databases/webview.db file contains highly confidential user information, such as the user's full first and last name, username, and the email account tied to the flickr account used on the device.

Another highly valuable database used by flickr resides at /data/data/com.htc.providers.uploads/uploads.db-wal. As seen in Figure 30, this database is a record of every upload attempted by a user and whether the upload was successful or not. It also contains the clear text data the user was attempting to upload with the photo.

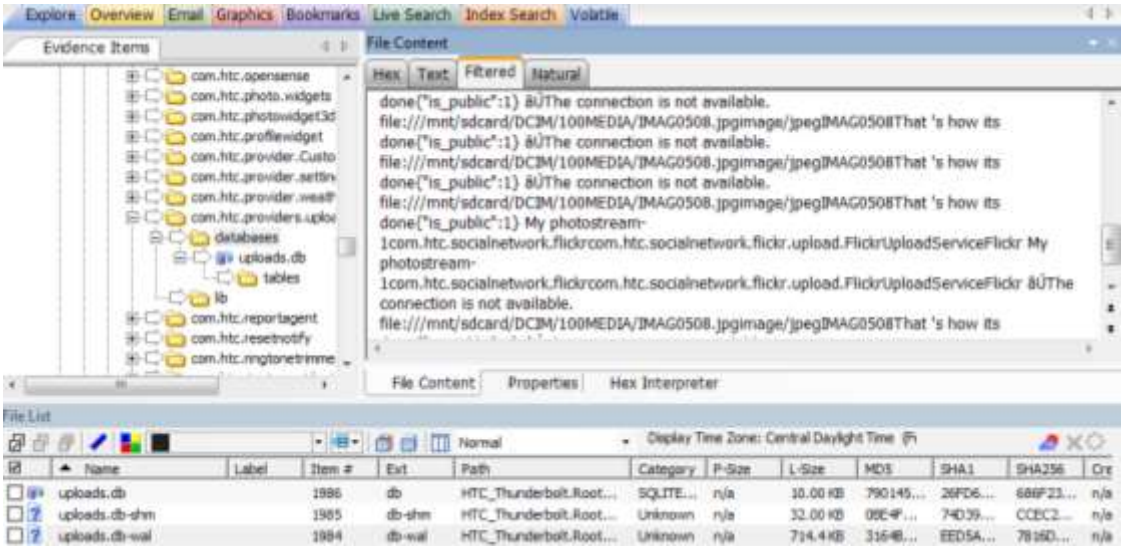

Figure 30: Flickr upload data from the /uploads.db-wal file (FTK)

Gmail

As discussed, Android platforms were built by Google. Users must have an active Gmail account to download applications from the Android market. Therefore, it can be assumed with high probability that an Android phone is linked with a Gmail account. Whether the device owner utilizes the account is completely up to their preferences and needs, however the Android platform is built around Google and strives to have their applications as the 'go-to' applications on the device.

If the Gmail application on the Android device is heavily utilized by the HTC device user, it may store a plethora of data. The /data/data/com.android.providers.contacts/ location discussed earlier contains a large amount of Gmail data, including full first names, last names, and email addresses of many of the contacts stored on the phone. It also displays the HTC user's Gmail groups. The /contacts2.db-wal file contains the phone number list for most of the contacts in the phone. Therefore, if a user was not able to utilize MPE+ for a forensic analysis however was still able to use another program for the logical analysis, they would still be able to pull the confidential contact names, emails, and phone numbers from the device by analyzing this folder.

The /data/data/com.google.android.gm/databases folder contains databases specific to each of the different Gmail accounts loaded onto the device (see Figure 31). For example, clicking on the /mailstore.mobartifacttest1@Gmail.com.db file outputs a database filled with data from the mobartifacttest1@Gmail.com inbox that was last accessed on the device. This data includes personal email addresses of the sender and recipient. It also may contain the subject lines and previews or portions of the email, as well. The /webviewCache.db file contains URL locations of graphics loaded by emails opened by the user. Though many of these graphics are created for mobile devices and do not carry much significance on their own, they can be used to prove the existence of emails in an account that may have been deleted.

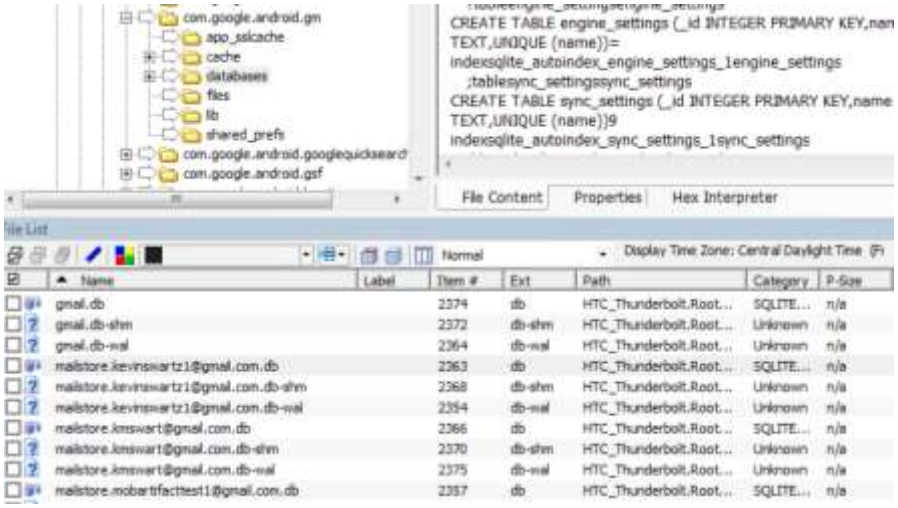

Figure 31: File contents of the /com.google.android.gm/databases folder (FTK)

Google Searches

Also under the /data/data/ folder is the location /com.android.browser/databases. This folder contains two extensive databases: browser.db and browser.db-wal. Browser.db contains full Google searches by the device user as well as any pages that the user has bookmarked. Figure 32 contains a side-by-side comparison of the data found in this file and a screenshot of the actual imaged device's bookmarked webpages. Browser.db-wal contains links clicked in emails, Google searches made by the user, email addresses from parties involved, and user email ID numbers.

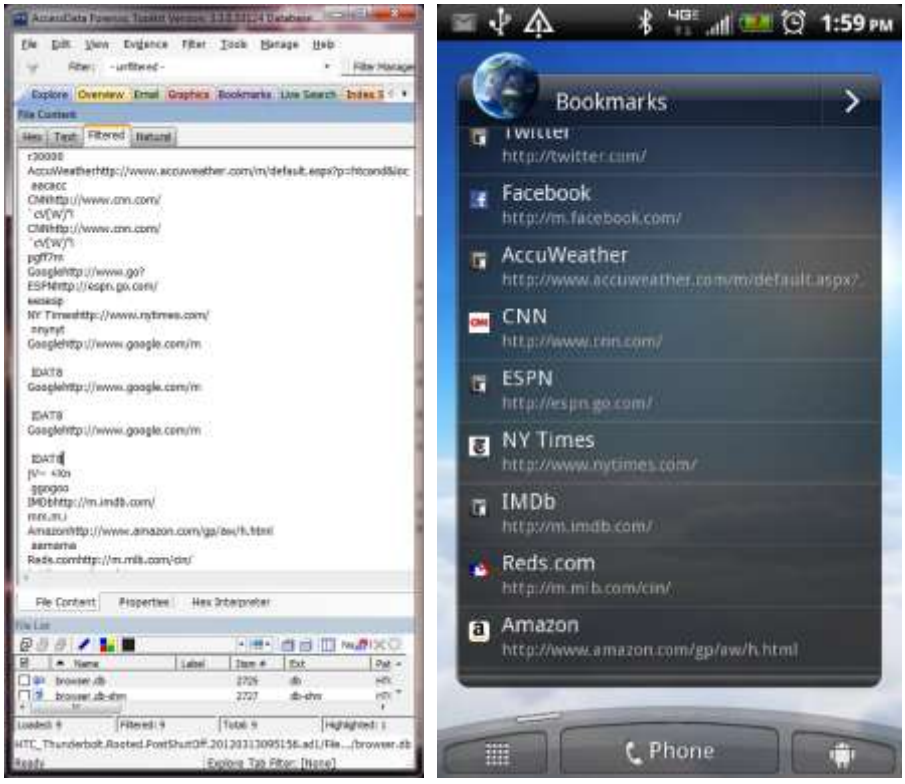

Figure 32: Bookmark data found in FTK (left) and the bookmarks app on the user's phone (right)

Phonebook

The previously mentioned /data/data/com.android.providers.contacts/Contacts2.db (and Contacts2.db-wal) files contain phone groups, contact names, and contact phone numbers. If the examiner had an issue viewing the data in MPE+ they can access this file to obtain the same data. If the 'data carve' option is set during pre-processing in FTK the image will also contain carved data from the contacts list. This can include contact's photos downloaded automatically from Facebook and linked to their corresponding accounts. Figure 33 contains an example of such a data carve.

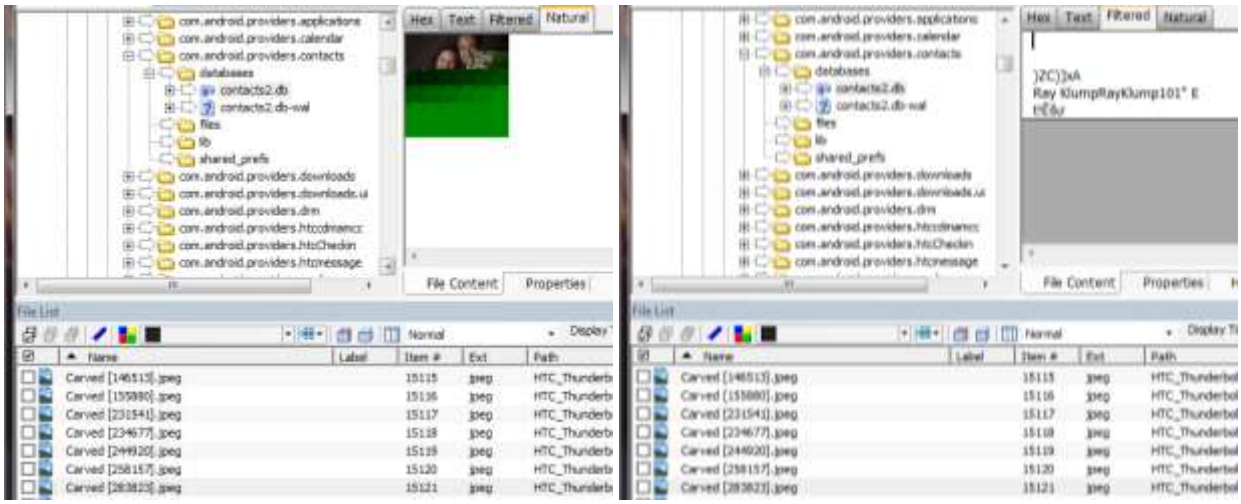

Figure 33: Carved photo (left) and the contact name (right) available in the Contacts2.db (FTK)

MMS

The images from MMS messages that are stored on the phone are located in the file /data/data/com.android.providers.telephony/app_parts.

Unless sent as such, these photos are not stored as thumbnails, as many photos found in databases are. Instead, these are full size, high resolution photos.

SMS

If the examiner was unable to view SMS messages during the initial examination with MPE+ they can find the same data in a few different locations on the phone. The data is found in the /data/data/com.htc.cs/databases/MWDB file, the /data/data/com.android.providers.telephony/databases/mmssms.db file, and the /data/data/com.htc.messagecs/databases/messagescs.db (and messagescs.db-wal) file.

Calendar

The default calendar app on an Android device stores its data in the /data/data/com.android.providers.calendar/calendar.db file. This file contains quite a few of the stored calendar items in clear text along with their corresponding date stamps. It also contains clear text of each calendar tied to the application, many of which are identified by either email address or the user's first and last name.

Other Applications:

Endomondo

Endomondo is a fitness application used by the device owner. It uses GPS to track a user's run, bike ride, or other outdoor workout to provide detailed feedback to the user. For example, if Endomondo is started at the beginning of a run and stopped at the end, it will output the total time, miles ran, average speed, and average mile time, among others. It also outputs a detailed map of the path travelled by the user.

While the application does a good job at encrypting GPS data and map data, the file /data/data/com.endomondo.android.pro/shared_prefs/com.endomondo.android.pro_preferences.xml contains the user's name, password, distance goals, and local paths near their typical running route (see Figure 34).

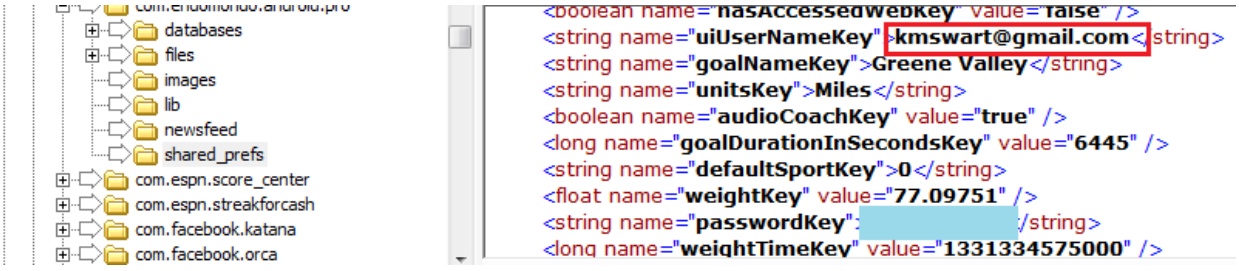

Figure 34: Username and password (covered for security) from the Endomondo application (FTK)

MLB At Bat

The user credentials and password used to login to the MLB At Bat app were discoverable in clear text on the device. In fact, as shown in Figure 35, the /data/data/com.bamnetworks.mobile.android.gamedy.atbat/_preferences.xml file labeled the username and password before writing them in full text. It also stated the user ID number and the expiration time of the user's current session. Another file of note in the same location is the /shared_prefs/checkinHistoryMapPref.xml. This file shows every 'check-in' the user has completed at a Major League Baseball stadium with the application. For example, if a fan goes to a Cincinnati Reds game at Great American Ball Park they can 'check-in' at the game from the MLB At Bat application. The check-ins are stored by date, location, and time.

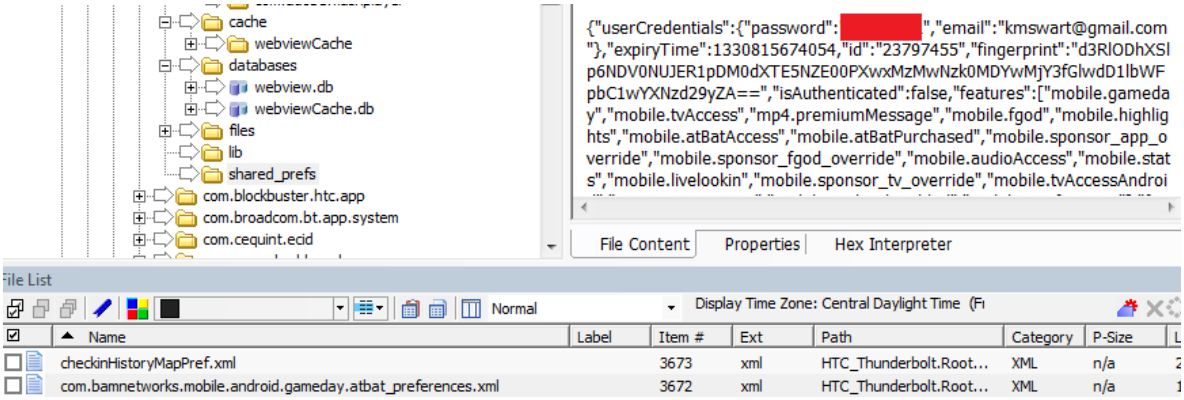

Figure 35: The username and password for the MLB At Bat application (FTK)

ESPN Score_Center

The ESPN ScoreCenter application stored on the target device detailed the username and password used to login to the application (and, subsequently, ESPN.com as well) in the /data/data/com.espn.score_center/databases/webview.db file. It also listed websites and articles the user has viewed, cookie information, and a user's self-identified favorite sports teams.

Google Maps

The Google Maps application stores the history of locations recently searched or travelled by the user while using Google Maps in the /data/data/com.google.android.apps.maps/databases/da_destination_history/ folder. For example, though the application had been much more heavily used in the past, there was only one location available in the folder on the tested HTC device. Nonetheless, the data available was the full address with street name, number, city, and zip code, all in clear text.

Youtube

Video previews on Youtube are typically thumbnails of a snapshot of the video. Any thumbnails accessed by the HTC device were location in the /data/data/com.google.android.youtube/cache file. These files were discovered by FTK during the data carving process.

Phone Locator Pro

Phone Locator is an application loaded onto the device by the user as a security feature. If the user loses their phone they are able to use this application to activate certain features on their phone to assist in locating it. For example, the user can remotely text the device a code and instruct it to email them the GPS coordinates of the phone if lost. They can do the same to start the phone ringing at the highest level to assist in finding it, lock the SIM card down, or wipe the phone and any SD cards entirely. If the phone has a front facing camera, it can also be programmed to take a photo with the camera facing the user if the login password, pin, or pattern is ever entered incorrectly. This photo is then immediately emailed to the user. This is an incredibly useful feature. If the user was ever uncertain if they had lost their phone, they could check their email account for such photos. If they have received an email from Phone Locator Pro with a photo of someone else attempting to log in to their phone, they would know their phone has been stolen. They could then turn on GPS tracking to assist in recovering their device.

While the application does a great job in protecting the phone from the average user, it provides a great deal of confidential information during a forensic analysis. For example, the /data/data/com.rvo.plpro/shared_prefs/com.rvo.plpro_preferences.xml file contains every SMS trigger (text messages that can be sent to the phone to remotely activate its features), the Gmail address of the user indicating where emails and logs are to be sent, as well as any phone numbers tied to the account. It also describes which of its features are active. This can be an indication for a forensic analyst that they should attempt to disable the application to avoid a possible remote wipe.

Other important evidence items discovered on the device:

Wi-Fi Data

Typically, Wi-Fi SSID names and passwords are highly confidential, especially for businesses intent on protecting data. As shown in Figure 36, the /data/misc/wifi/supplicant.conf file discloses any

network SSID names and encryption keys stored on the phone in clear text. If the attacker has this data they can authenticate with any network listed, as long as they can find it and connect to it.

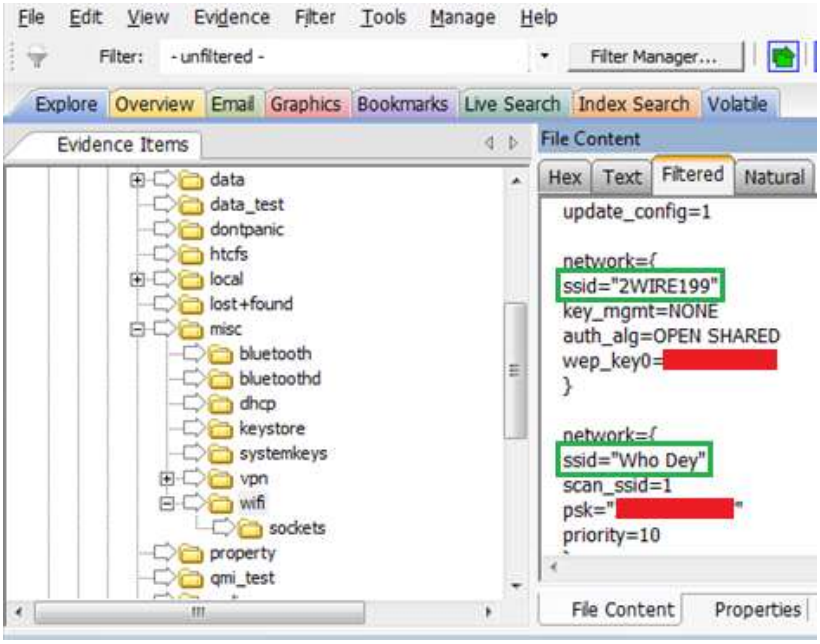

Figure 36: Wi-Fi SSID's and their corresponding keys stored on the phone (FTK)

Other Confidential User Data

Another file, located at /data/system/accounts.db, contains usernames and emails for most of the accounts and applications tied to the phone. It also contains a username and password taken from the /com.htc.cs folder in clear text. This password authenticates the user to their Gmail account. It can be seen in

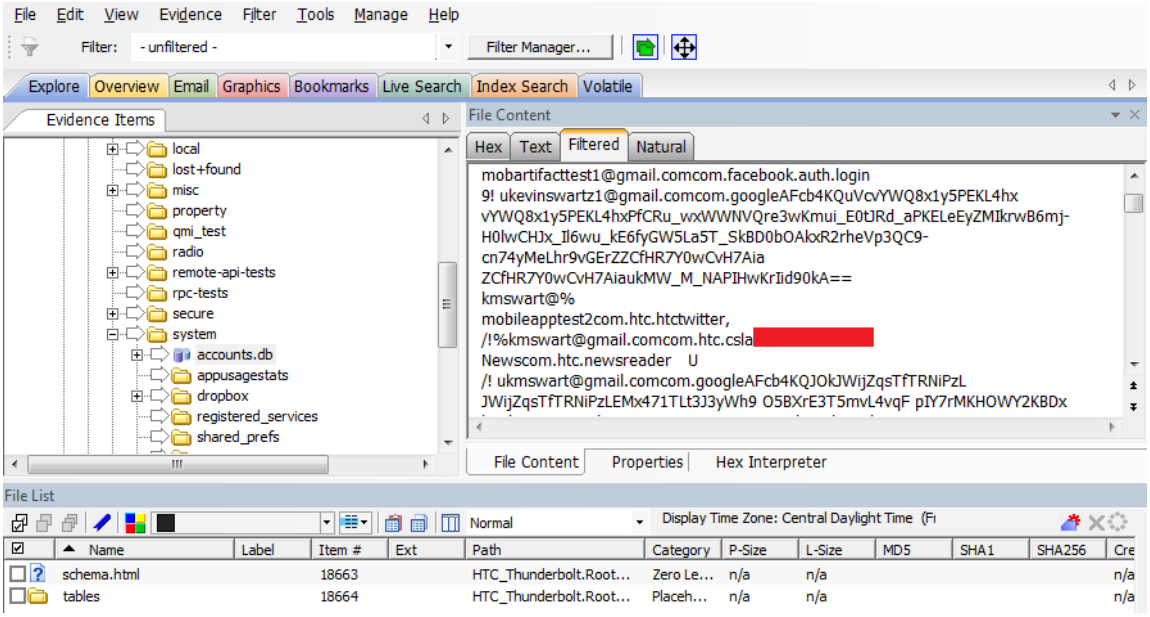

Figure 37 (with most of the password covered for security).

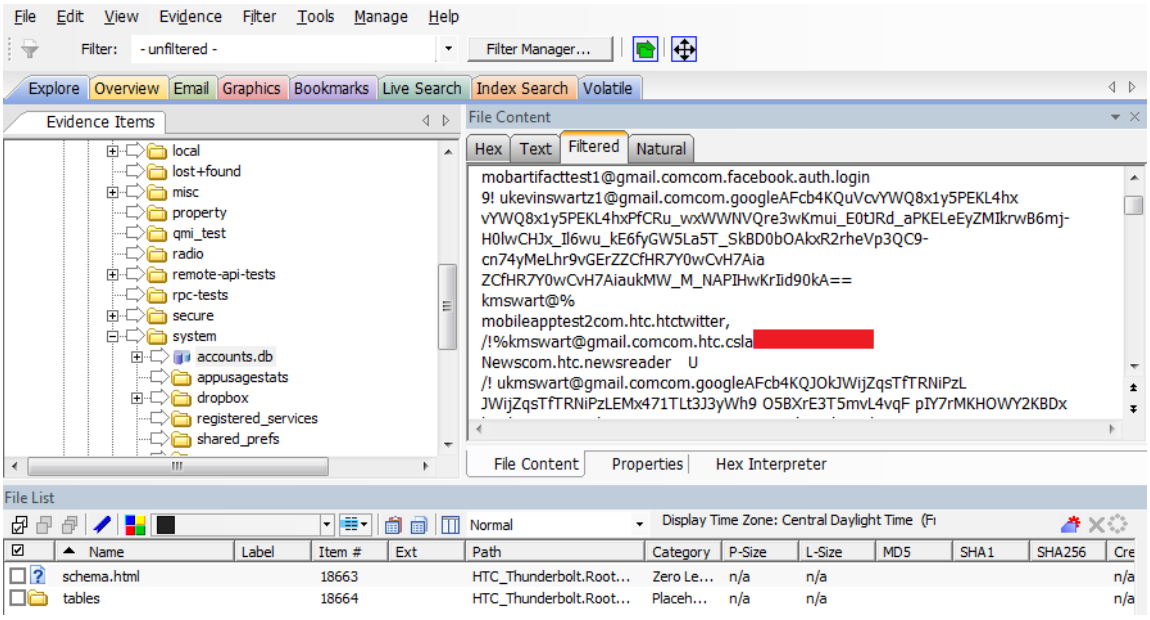

Figure 37: Gmail username and password available in clear text (FTK)

The data/data/com.android.htccontacts/shared_prefs file contains account types, websites, and email addresses associated with every account on the device.

The/data/data/com.htc.cs/shared_prefs/ folder is a storage place for a large amount of confidential user data. For example, as seen in Figure 38 the /CSPreference.xml file contains the user name and phone number, as well as the clear text "City born in" followed by the city the user was born in. This is most likely a security question and answer and may vary from device to device depending on what question and answer was input by the user. The /CSShared file contains the user's hashed password, full first and last name, and email information in the clear.

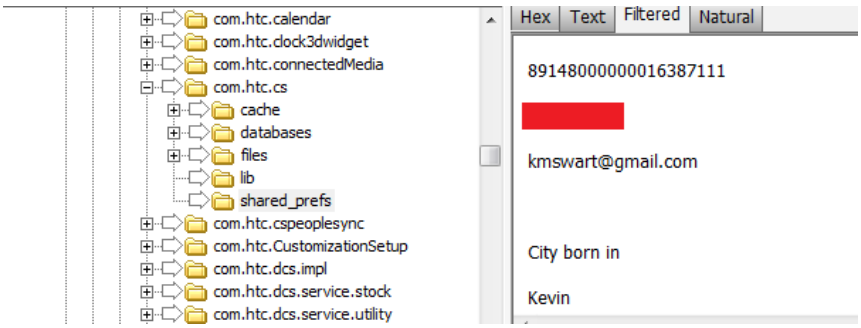

Figure 38: Email address, name, and security question and answer (answer covered) (FTK)

All media mounted on the device's external SD card is listed in the file /data/data/com.android.providers.media/databases/external-258ffa3.db. Most of the media accessed by the device or stored internally is listed in the /internal.db file at the same location.

The phone number of the device, what type of network it runs on, and the user's allotted monthly data from Verizon can be found in the /data/data/com.vzw.hss.myverizon/shared_prefs-DataMeterWidgetPrefs.xml file.

Step 6: Reporting and Recreating Duplicate Results

Issues in Rooting

If the purpose of a particular forensic analysis is to bring actionable evidence to a company, employee, or legal case, a competent and trained forensic analyst must be able to duplicate the results that the original analyst obtained with as few changes to the system as possible. This is where 'rooting' becomes an issue with Android devices. When a device is rooted it undergoes drastic changes to its OS. Many of the security features are circumvented, permissions are granted where they normally would not be, and in many cases an entirely new OS is loaded onto a device. In this instance it is critical that every step is documented. It is also crucial that every step be indisputable as to what happened to the device. Otherwise, the opposition in a legal case may be able to find faults on inconsistencies in the forensic analysis of the device and have it dismissed from court. For example, they could imply that the forensic examiner added the data in question to the phone while making changes to the device. They could also infer that data was added or deleted via the network if the device was not handled properly by being shielded from the network as soon as it was discovered.

Since rooting a device may be necessary to obtain the most crucial data on a device, a forensic analyst may find themselves in the unpleasant situation of having to root a device before examining it. To do so, however, they must first ensure that any rooting procedure will not erase all original data on the device. In this study, the device's firmware had to be downgraded effectively deleting all critical data from the device and rendering the forensic analysis on the rooted device useless.

Issues with Setting the Phone in 'Disk Drive' Mode for Physical Analysis

Another issue with Android devices is the requirement that the device be in a specific state or mode for an effective forensic analysis to be completed. Taking a physical analysis of a device is simple enough if the device uses a standalone SD card, however a built in NAND flash system that is segmented as an on board SD card can be problematic for an examiner. As discussed earlier, to obtain a complete physical analysis of an onboard SD drive or a physical analysis of a device before removing the battery, the device must be set into its 'Mount as Disk Drive' mode. To do this the device must be accessed and the modification must be made as one of the options on the phone. If the phone is password protected the password must be circumvented before the examiner is able to enable this mode. Even after accessing the phone the examiner must make modifications to the device, altering its original state and bringing its validity as evidence into question.

Apple iOS Forensics

Step 1: Deciphering the OS and its File Structure

While there are many differences between the two types of devices – Android and iPad – there are just as many similarities. Both devices are loosely based off of a Unix or Linux data structure. Just like an Android device, Apple utilizes NAND flash memory as the preferred on board memory of an iOS device. While many Android devices have space for external SD memory cards to be added and utilized as additional storage space, however, Apple devices such as the iPad and iPhone do not. All data is stored on the built-in NAND flash in these devices. [7] The file structure, while similar to a Linux file system, is proprietary to Apple. Known as the Hierarchical File System Plus, or HFS Plus, the file system is based on the Mac OS file system previously developed by Apple.

The key difference between the two devices is the use of an SD card and its effect on file structure. While an iOS device will only have on board memory and an HFS Plus structure, the Android device may use two separate file structures: the Linux based file structure for the on board memory and NTFS file structure for any removable memory and media. IPad devices, like Android devices, use flash memory (Random Access Memory, or RAM) as their volatile memory of choice. According to Andrew Hoog and Katie Strzempka, RAM can contain extremely valuable forensic data, such as passwords, encryption keys, user names, app data, and others. [7] When a device is turned off, any data that was stored on this memory is lost. At the time of this writing there is no known way to acquire live data from the RAM on an iOS device.

Just as it is on an Android device, data is 'sandboxed' on an iOS device. This means that applications are only allowed to access the data to which they have been given explicit permission. Like on the Android, this feature is an added level of security for both Apple and the user. It shields the file structure from unauthorized access by applications while also separating applications from user data which they should not have access. [17]

There are two main disk partitions on an iOS device. The first is a firmware partition that handles upgrades to the device. It also stores the OS and basic applications. [7] The second is the user data partition. This partition is where a majority of the valuable forensic data to be acquired is found, and will be a focus of this study. [7]

Step 2: Data Preservation on an iPad Device

Many of the threat types that are present for a proper forensic examination of Android devices are also present for iOS devices. For example, the risk of a network wipe is just as possible with an iOS device as it is with an Android device. In fact, the threat may even be greater with iOS devices. This is because iPad and iPhone devices come loaded with the MobileMe software suite. The software allows a user to remotely wipe the system [17], whereas Android users have to install third party software to have the same functionality.

As discussed earlier, to prevent against a remote wipe the device must be protected against all network access. This can be achieved by placing the device in airplane mode, turning off all radios on the device, or placing the device in a network proof container. The issues with these methods are similar to the issues with the same changes on an Android device. For example, accessing the device to turn off radios makes changes to the device leaving the possibility that those changes could be challenged in a court of law. Also, if the device is locked it may be extremely difficult to circumvent the password (if the password is not known). There are device settings that allow the device to be wiped if it has not been accessed after a certain period of time or if a certain amount of incorrect passcodes are entered (the latter protecting against a brute force attack against the password), wiping the device clean of crucial evidence and rendering a successful forensic analysis as improbable.

Apple is a strong proponent of device security and has implemented many security features that affect business users and forensic examiners. While many of the features will be discussed in depth later in this study, there is one key feature of note that directly affects data analysis. According to Apple, both iPhone and iPad devices offer hardware based encryption. They have added a 256-bit AES encryption to protect all of the data stored on a device. [17] This can be troublesome for a forensic examiner because, if implemented, it would make most of the recovered data unreadable. This feature is a valuable asset to business owners that want to use tablets and mobile devices while preserving data confidentiality and security.

Step 3: The Forensic Analysis of an iPad Device

Physical and Logical Analysis

Using the same forensics tool commercially available through AccessData, the Mobile Phone Examiner Plus (MPE+), full logical and physical analysis of iOS devices are possible. While there are some instances with iOS devices that require much more invasive methods of acquisition, the examinations completed using MPE+ for this study proved to be highly efficient and effective.

Installing Required Software

A complete forensic examination of an iOS device requires that any device drivers used by the target device as well as a copy of iTunes are installed on the host system. QuickTime should be uninstalled from the host device as well. Before analysis is completed, the auto sync mode in iTunes must be disabled. [18] If left enabled, the iOS device may sync with the host iTunes, deleting or modifying critical data and rendering a sound legal analysis of a device unlikely.

Connecting the Device and Completing the Examination

IOS compatible USB cables should be used to connect both the target device and the host system being used to examine it. After the connection is made, the user needs to load MPE+ and select the appropriate manufacturer and model for their examination. From there, the user should select the "Full Disk" option to obtain the user partition, the OS partition, and any slack space on the device. The user should then choose where to save the AD1 file. If all options are acceptable to the examiner, they only need to click the "Finish" option to tell MPE+ to begin the examination. [18] MPE+ is a powerful tool to use in Apple examinations and proved to be extremely efficient when doing a full physical and logical analysis of the test device. For this study an Apple iPad was examined and, as such, the procedures and results referenced in this study represent that platform.

Step 4: Data Recovery

Similar to the Android system, data carving is an efficient way to obtain data that was previously deleted on a device, such as GPS coordinates, photos, videos, and more. MPE+ will extract the data carved from a device, however it will not be viewable by an examiner until it is loaded into the Forensic Tool Kit and separately carved. As with the Android device, during case pre-processing specific focus should be paid to data carving options and as many carving options should be selected as possible. While loading the forensic image into FTK, the examiner should instruct the Forensic Toolkit to data and meta carve, ensuring that as much data can be extracted from the target device as possible. This will allow the examiner to discover 'deleted' data that may have been only partially overwritten by the operating system as well as data that has been hidden in unallocated space.

Step 5: Analyzing the Results

Analyzing the Data from the Physical and Logical Image Using MPE+

As discussed, the MPE+ tool can take a full physical and logical image of an iOS device. This is in stark contrast to the Android platform which requires multiple modifications to be made to the device before a logical analysis can be completed.

While the two platforms have different file systems, the hierarchical structure of the iPad's file structure is similar to the file structure found on the Android. Instead of the /data/data/ folder that

contained most of the confidential data was found on the Android, the iOS device stores most logical data under the /private/var/ folder. Additionally, most mobile applications used by the device store their data under the aptly named /private/var/mobile/applications folder. As with the Android device, while the Mobile Phone Examiner Plus is a great tool for acquisition, it does not allow the user to go into as much detail as the Forensics Tool Kit. It does allow the user to see similar information as the Android device such as contacts, SMS and MMS messages, call logs, and the file system, along with corresponding files discovered during the examination. The user should then utilize another toolkit such as FTK to analyze these files more closely.

Analyzing the Data from the Physical and Logical Image Using FTK

iOS Analysis: The Evidence Tree

The physical data extracted from an iOS device is stored under the individual file system folder in the evidence tree. For example, the DCIM folder contains all photos that were stored on the device and extracted during the physical examination of the device. While there are many other examples of physical data on a mobile device such as contacts, videos, and others, the device imaged did not contain any of that information. If the reader is interested in learning more about iOS physical forensics, it is highly recommended they look to the bibliography for some wonderful resources about the subject. Unless otherwise noted, for path references below the full path in the evidence tree is /[device name]/File System/private/var/mobile/applications/ (see Figure 39).

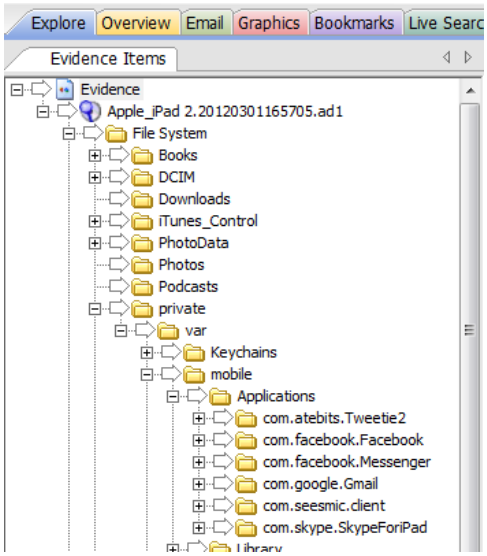

Figure 39: The iPad evidence tree (FTK)

Discovering Artifacts by Application:

Facebook

The Facebook folder is easily identifiable on an iOS device. It resides under the appropriately named /com.facebook.facebook/library folder. There are a few files of importance in this folder. As seen in Figure 40 the /preferences/com.facebook.facebook.plist file contains the user's Facebook username, email, user ID, full name, and links to their profile. The /com.facebook.Messenger/library/preferences file contains similar information while also including the last time the Facebook Messenger application was updated and the user's Messenger user ID.

Figure 40: The facebook.plist file (FTK)

Twitter

The Twitter application loaded on the device stored the user name in clear text in the /com.atebits.Tweetie2/library/preferences/com.atebits.Tweetie2.plist file. Other confidential Twitter data stored specifically by the Twitter application was not discernible in this or any other file.

Seesmic

Seesmic stores its data in a very similar location on the iPad as it did on the HTC Android device at /com.seesmic.client. There are two locations to pay attention to in this location: the /documents folder and the /library folder.

The files in the /documents location contain full user photos, user ID numbers, HTML links to profile pictures, and entire conversations and messages sent on both Facebook and Twitter in clear text (see Figure 41). The /library location contains the file /mmsdk/mmhandshake.archive that details the last handshake made in Seesmic along with time stamps detailing when it is going to attempt the next server handshake.

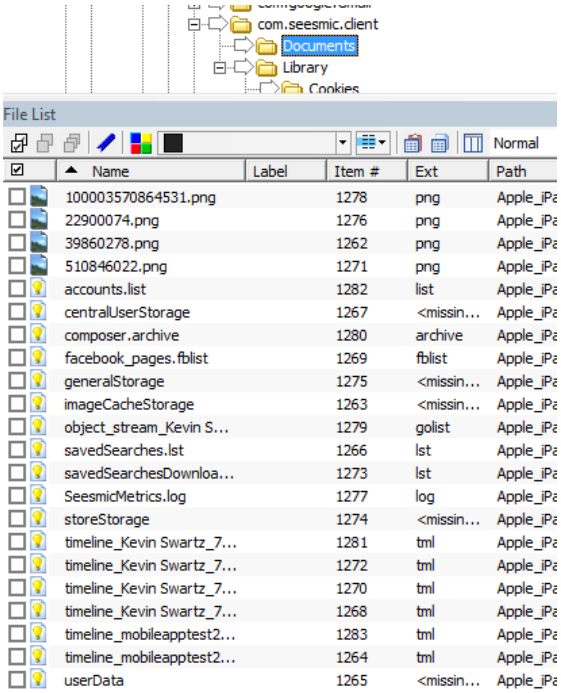

Figure 41: The /com.seesmic.client/Documents folder and its contents (FTK)

Skype

The /com.skype.skypeforipad/library/applicationsupport/skype/mobapptest3 (the 'mobapptest3' is the username of the Skype account on the device) folder contains valuable files for a forensic examiner. For example, as seen in Figure 42 the /chatsync file contains the Skype user's text chats in clear text with usernames of all parties involved. The /main.db/schema.html file contains full chats and user ID's. It also details computer and camera settings during any voice or video calls made with the service.

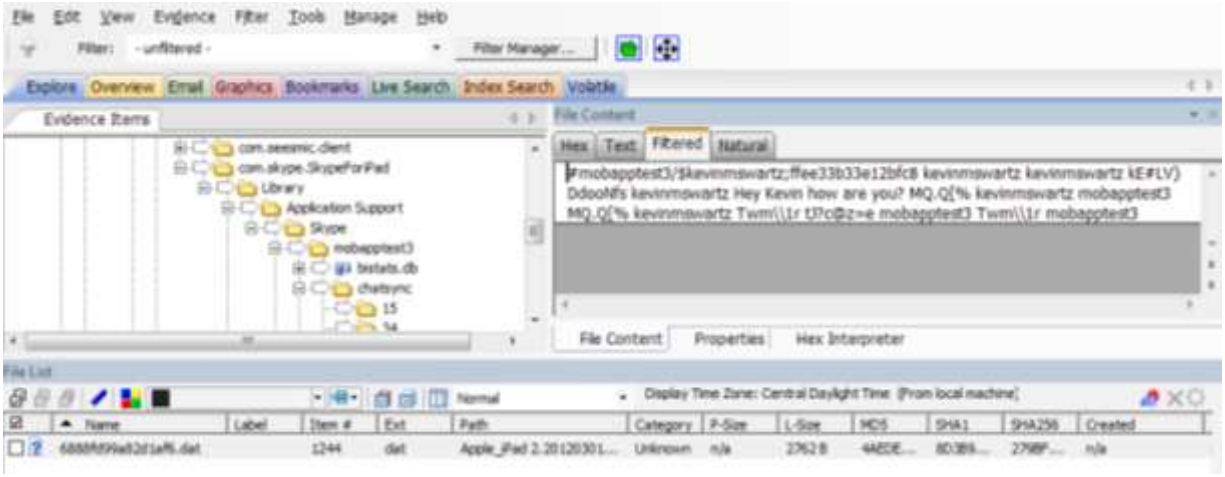

Figure 42: Usernames and clear text chat over Skype (FTK)

Also under /main.db is the /tables folder which contains a majority of the Skype user's information. For example, the /messages folder contains data on all contacts made, whether they were voice calls, video calls, or chats, as well as the parties that were involved and any other data related to the chat. The /conversations file contains a database filled with timestamps that detail the user's recent activities and inbox data. The /chats folder contains full chats with usernames. The /contacts folder contains a detailed contact database with the user's Skype names, birthdays, identified languages, genders, cities, phone numbers, links to profile photos, and buddy list numbers (most of this data can be seen in the screenshot in Figure 43. In the figure the text "19110101" is actually the user's birthday as it was input when they setup their Skype account (January 1, 1911).

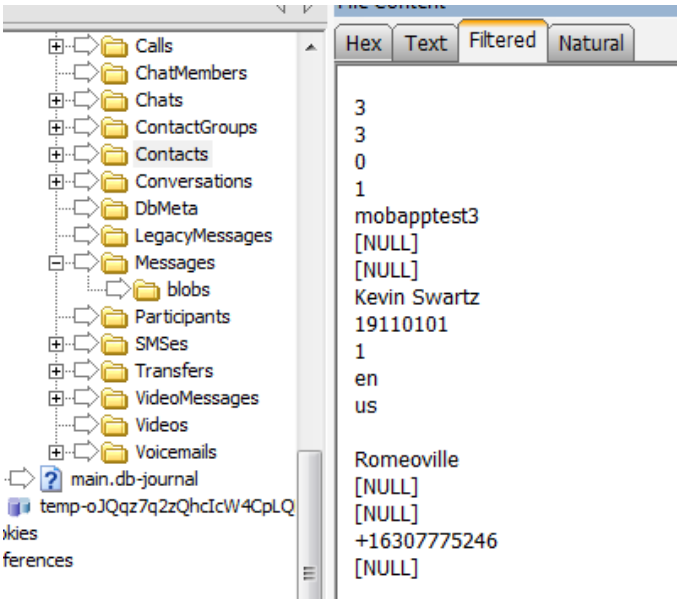

Figure 43: Data recovered from the skype/[username]/contacts folder (FTK)

Gmail

The Gmail application downloaded onto the test iPad device stored its data in the /com.google.gmail/ folder. A majority of the inbox was available in clear text by viewing the /library/webkit/databases/https_mail.google.com_0/0000000000000001.db file. The data in this file details whether the email has been opened, conversation ID's, subject lines, and email addresses of parties involved. Full emails are available in clear text near the bottom of the database (see Figure 44). Similar data can be found in other locations as well, such as the /tables/cached_messages folder.

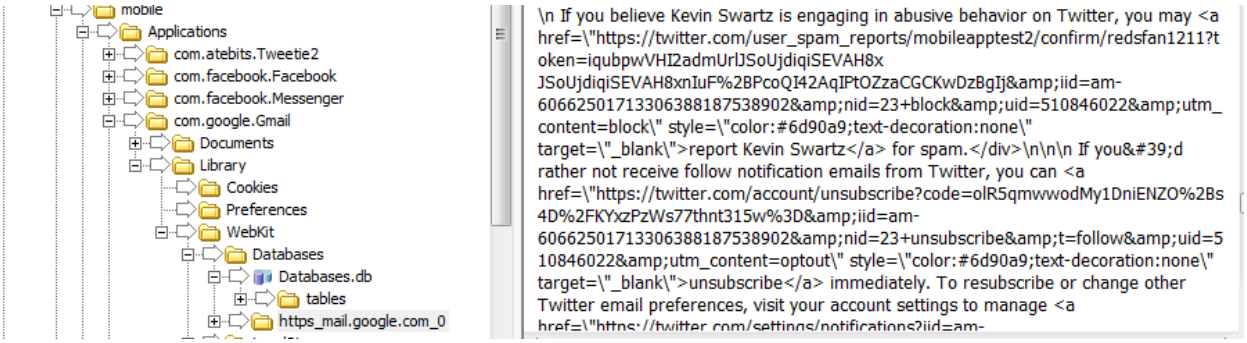

Figure 44: A clear text email from the user's Gmail account with HTML coding (FTK)

Safari

The /private/var/mobile/library/safari/history.plist file contains information about websites that were connected to on the iOS device using Safari, Apple's native browser. For example, one of the websites visited by the iPad's user was a password recovery webpage from Apple. While it does not include the user's password in the forensically recovered file, it does include their username and other identifiable data.

Other important evidence items discovered on the device:

The /mobile/media folder contains multiple files detailing all DCIM photos stored on the device. It also contains copies of each of these photos.

The /preferences/systemconfiguration/ folder contains multiple files, including com.apple.network.identification.plist, com.apple.wifi.plist, and preferences.plist, that contain detailed information about wifi connections stored on the device. For example, the network.identification.plist file contains the internal IP address and MAC address for the router the device was connected to. The mobilegestalt.plist states the "UserAssignedDeviceName" (Lewis 13106), while the prefereneces.plist contains device wifi preferences, such as the preferred network (WIN@LEWIS – see Figure 45).

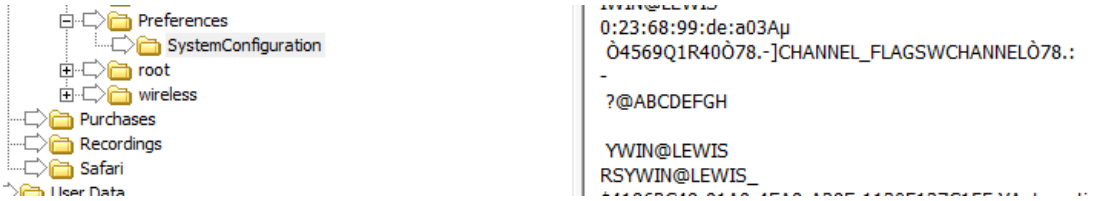

Figure 45: Stored network information from the iPad test device (FTK)

Since the iOS device tested was an iPad, there were no traditional ways to make voice calls or send SMS and MMS messages from the device. The device still contained the Call_history.db file in the /wireless/library/callhistory which would have contained the call history were there any calls made on the device.

Step 6: Reporting and Recreating Duplicate Results

While this study did require the use of iTunes to connect to the target device, there was no major system changes required to obtain the targeted data. As discussed earlier, the file system is divided up into two disk partitions – the first houses the OS and basic applications while the second contains a majority of user data. Any upgrades to the device firmware will only affect the partition on the device that houses the OS data. This results in the user data remaining entirely untouched after an upgrade or downgrade. Therefore, if an examiner needs to upgrade or downgrade a system to be able to complete a forensic analysis of a device they would be able to do so without the threat of this action deleting critical data.

Hands-On Examination of Android and iOS: What Was Learned

In all, six different email addresses were found in over twenty-five different files combined that were stored on the two devices. Over 3,000 photos and thumbnails were discovered. Every user ID and user name of the social media applications stored on the devices was found. A strong majority of user contacts, including social media and email contacts, were discovered. The Android's full contact list with phone numbers, social media user ID's, and email addresses were detailed, as well. Websites viewed by browsers were available in multiple locations, as were user's Google searches. GPS and map locations were found on the Android device.

Multiple files were discovered as well that, if lost, could cause immediate security concerns. Every Wi-Fi SSID and password stored on each device was discovered. Four different user passwords were found in a total of six different files alongside their subsequent usernames, giving an attacker all they need to login to any one of the accounts. If the hacker wanted to attack another account of the

user but did not have the correct password they could use one of the user's security question challenge/answer strings discovered on the devices. Full Twitter and Facebook feeds as well as confidential email inbox data were discovered on both devices. SMS and MMS messages were found on the Android device including detailed information about each, such as who the messages were to, who they were from, and whether they had been opened or not. The user's calendars were discovered on the Android device and included dates and times of past and future events as well as the stored information about the event in clear text. Authorization tokens to multiple applications, social media or otherwise, were discovered on devices.

Overall, a great deal of confidential, unencrypted data was available on both devices. While the Android device required modification before the data could be retrieved, the device in itself provided much more clear text information than the iOS device. Even though the Android had a much greater amount of data stored on it due to a much greater amount of use than the tested iPad, when the data was analyzed application data that was available in multiple locations on the Android device was much less commonly found on the iPad. While both systems allow applications to run only in a sandbox environment, it seemed that the iOS platform was better equipped to reduce the amount of confidential data available in clear text on the device.

How Artifacts May Be Used Against Original Owners

While reviewing what can be discovered on a device, it is important to grasp the consequences of lost data as well as how it can be used against its original owner. Photographs and contact lists may not be devastating to a traditional personal mobile phone user (other than the understandable heartache of losing any data that is not backed up). The loss of such data, however, can be catastrophic for businesses. Messages and emails are confidential and, in an ideal world, would stay that way. Unfortunately, there are attackers that would love nothing more than to have that data (if nothing else just to say they could get it). Usernames, passwords, authorization keys and session tokens can be used in a variety of ways against a business or individual**.**

Attackers may use data discovered on devices to initiate a 'traditional' attack on information security. According to Michael E. Whitman and Herbert J. Mattord, "an attack is an act that takes advantage of a vulnerability to compromise a controlled system." [19] They describe some of the most common attacks including malicious code, dictionary password attacks, spoofing, social engineering, man-in-the-middle attacks and packet sniffing, among others. Each of these attacks can be performed with the data discovered in this study with a high degree of efficiency.

Social engineering is "the process of using social skills to convince people to reveal access credentials or other valuable information to the attacker." [19] Infamous hacker and social engineer Kevin Mitnick has been quoted as saying "People are the weakest link. You can have the best technology; firewalls, intrusion-detection systems, biometric devices… and somebody can call an unsuspecting employee. That's all she wrote, baby. They [just] got everything." [20] While social engineering will be discussed in depth later in this study, it is important to understand, as it can (and typically will) be used in conjunction with other attacks to make the attack itself as successful as possible. The following sections provide examples of how to combine forensic examination with well-known attacks.

Malicious Code

A malicious code attack involves the attacker executing a virus, worm, or Trojan horse on a system. This can be done by the attacker themselves or unknowingly by a user. For example, by using

the data found in the /data/system/accounts.db file, an attacker may be able to login to a user's Facebook account with one of the usernames and passwords discovered in the file. From there, the attacker could post a link that, when clicked, executes malicious code on the system of whomever clicked the link. It is traditionally more believable for a user to click a link from someone they know and trust than from a total stranger. With that in mind, if an attacker has the ability to assume another person's identity in any way to assist them in carrying out an attack they traditionally will not let the opportunity pass.

Malicious code could also be 'pushed' from an online source. For example, both devices allow users to download apps online. In many cases, once the application is purchased online it will immediately be sent to the phone, downloaded and installed. While Apple has strict control over the content of their app store, Android does not. Malicious programs have found their way onto the 'Google Play' market. If an attacker obtains a target's user name and password they could log in to the user's Google account and push a malicious program to the unsuspecting user's phone. This app could be programmed for a variety of attacks, including obtaining personal data or redirecting user calls to a premium call hotline.

Dictionary Password Attacks

There are many different ways to attack a password. Downloadable applications such as John the Ripper can be used to attempt to crack passwords by instructing the computer to use a list of passwords against a target. The Password Recovery Tool Kit (PRTK) that is part of the FTK suite of tools can be used in a similar way. A user can load word lists or use a pre-installed list to attempt to crack the password of a file loaded into the application. These programs can be extremely efficient in cracking passwords if the list that is fed into them contains multiple passwords previously used by the owner of the file being cracked. For example, had the user of the Android device password protected the .pdf files discovered on the phone, the file could be extracted from the image and imported into PRTK. From there the attacker would add any and all passwords discovered during the forensic analysis along with a dictionary list downloaded from FTK that contains all the words found on the Android device. They could then use that list to attempt to crack the password. If the .pdf file in this example contained confidential information, the data contained in the file could be uncovered even with the .pdf being password protected.

Attackers could use data found during the analysis to attack various websites, as well. According to Aaron Marcus, almost 60% of all users use five passwords or less. [21] If the attacker is able to gain access to six passwords, the same number discovered during this study, the odds are significantly in their favor for having the correct password for a website. For example, an attacker may try to gain access to a bank account using the usernames and passwords discovered on the HTC or iOS device. They could attempt to get into any site they believe the user has access to including each of the email addresses and social media websites discovered on the device.

A user's security question challenge and answer were found in the data/data/com.htc.cs/shared_prefs/ folder. An attacker can use this information to recover and change a user's password. They may do this to gain access to a unique account or to create a denial of service attack against the user by locking the user out from their own services. They can do this through the 'reset your password' option.

Typically, password resets are accomplished by inputting answers to security question challenges. If the user has this data, they can request the password reset, effectively gaining access to the user's account as well as denying a user further access to their own account. The attacker could

repeat this attack until the target account information is discovered or until they have a reason to no longer use it.

This type of attack could be completed to attack any account the attacker deems fit. For example, this could be executed against the user's Facebook, Twitter, or Gmail account, as well as the user's bank accounts, credit card accounts, work email, and VPN accounts among others.

Man-in-the-Middle Attacks

The man-in-the-middle attack, also known as the MITM attack, is an attack where "an attacker monitors packets from the network, modifies them, and inserts them back into the network." [19] This could be completed in multiple ways with the data obtained from the forensic examination. For example, since the MAC and IP addresses of the router that the iPad was connected to were discovered in the network.identification.plist file, an attacker could attempt to impersonate the wireless router. They would accomplish this by sending signals indicating that their system is actually the router that the target's system is looking for. The next time the user's device attempts to connect to that unique SSID and MAC address, it would find and connect to the attacker's device instead of the target router. While the attacker may simply pass the information sent from the victim to the final destination (either the internet or another node on the internal network), they would have unrestricted access to all the data flowing through the device. This could include usernames and passwords, confidential documents, and more.

Packet Sniffing

Packet sniffing is a passive form of a MITM attack. It involves watching (and at times recording) packets travelling over a network. Unencrypted data can contain highly confidential information, much like what was discovered as physical and logical data in this study. For packet sniffing to be successful, the attacker must have access to the target network. Since network SSID's, IP addresses, and passkeys were available in clear text in this study, an attacker with the information gleaned from a forensic analysis of these mobile devices and knowledge of where the individual networks were located would have no trouble pulling off this attack.

Spoofing

Spoofing involves acting like a trusted source to gain unauthorized access to computers. For this to work the attacker needs detailed information about the internal computer network. Since the mobile devices stored wifi data in clear text in the /data/misc/wifi/supplicant.conf file, the attacker would be able to modify their IP address to that of a computer cleared for access on an internal network to subsequently gain access to that internal network.

Social engineering

As previously discussed, social engineering is one of the most underrated and overlooked security fallacies in an organization. People are naturally trusting. That trust can severely undermine security, especially in a technologically advanced society such as ours. For example, some of the data recovered were photos with GPS coordinates. If the attacker found photos of a vacation with the GPS coordinates pointing to a popular vacation spot, the attacker could search through Facebook profile pictures discovered in the /data/data/com.facebook.orca/databases folder or the /data/data/com.android.providers.contacts folder, among others, until they find matches to each of the individuals in the photos. They could link the profile photo to the name of the individual using the

contact names, photos, and user ID numbers discovered in the Contacts2.db file discussed earlier. One simple way for an attacker to do this is to use the user ID number and add it to the end of the string: www.facebook.com/profile.php?id=. Plugging the resulting string into an internet explorer address bar would bring up the target individual's Facebook page (if public). This would provide the attacker with an even greater amount of actionable information such as the user's friends, birthdays, and more. If the profile is private, the attacker could attempt to access the original victim's Facebook account to gain access to the rest of the required information. They may also access text messages, phone numbers, Skype conversations, Twitter posts, and other data that would assist them with their attack.

They could use this information to approach their target and start a conversation with "I remember you from (insert fake chance meeting based on real locations that the targeted victim was at) with (insert friend or friends' names that were with the target at that time)." With this information the attacker has created a significant backstory that may allow them to socially engineer their way into significant, confidential data from one of the targeted victims.

This is one situation among many where social engineering can be used to gain access to confidential information. The best defense, as will be discussed, is to use precaution and never give out personal information unless you are absolutely certain that it is safe for you to do so.

Techniques to Prevent Against Attacks

Encryption Programs

As of the writing of this study, there are no full disk encryption programs available for an Android device. This means that if a device is lost the data will be in clear text when recovered. There are companies, such as WhisperSystems, that are attempting to create a full disk encryption for Android; however none are available on the market.

Apple has implemented encryption on their iOS platform for data stored on the NAND flash. In an article from the viaForensics website author Ted E states "Technically iOS encryption does work – the data-at-rest on the device is encrypted using a hardware encryption chip. It implements per-file keys that make deleted data recovery very difficult, and enables near "instant wipe" by deleting these keys. However, nearly all files encrypted on iOS can be decrypted even if the device has a passcode." [22] Therefore, while the iOS platforms do offer full disk encryption, there is a strong possibility that the passwords used to encrypt the data are able to be cracked. Therefore they should not be trusted as the only form of security on the device. This was not studied during this examination because the iPad used in the study was built before the encryption was implemented by Apple.

Password Policies

While there are multiple policies that should be implemented, password policies should be one of the first, most detailed, and most consistent. As discussed earlier, many of the forensics procedures require accessing the device to enable certain features. Adding a password to a device can be a simple way to prevent against access to a device, in turn denying the attacker the ability to enable the features and preventing a sound forensic analysis. In addition, MobileMe for iOS devices and third party software for Android allow users to set a device to automatically wipe itself after a certain number of incorrect login attempts, further protecting confidential data.

How to Protect Yourself and Your Organization

The Importance of Risk Management and Mitigation

This study has focused primarily on the procedures to obtain data from a mobile device as well as the artifacts left on those devices by social media. Since the data recovered can provide attackers with an incredible amount of actionable information they can use to attack an organization, it is important for business owners to understand the varying levels of risk mitigation when it comes to information security. While no organization is wholly secure from an attack, each organization has a responsibility to weigh the costs of deterring attackers (securing their data) versus the probable losses sustained by a successful attack.

According to CDW and MarketWatch.com, since 2010 one in four organizations has experienced some form of data loss. Companies with the most sensitive information are also the most highly targeted organizations. These companies reside in the financial services, health care, and higher education industries. Data loss in 2011 cost organizations in these industries an average of $5.5 million per breach. [23] CDW's study found that organizations that gave themselves an "A" grade in data security in 2011 were "more likely than others to require employee-owned mobile devices to comply with defined security procedures before they are granted network access". These procedures and their subsequent policies should include the ability to wipe a device by both the employee and the employer. If a company believes that a device has been compromised, having the ability to wipe a device provides both parties with a significant amount of protection.

While creating and enforcing strong policy is important to a company's overall security goals, it is not all encompassing. Verizon recently reported that ninety-seven percent of the 855 breaches they studied in 2011 could have been avoided through simple or immediate security controls. A majority of these breaches (612 out of the 855) occurred against small businesses with fewer than 100 employees. Many of the breaches were searching for businesses with remote access services combined with weak passwords. [24] As discussed, the acquisition of a user's mobile device by an attacker would provide the attacker with a large amount of data that could be used in an attack against an organization, significantly enhancing the probability of a successful attack. This would increase the probability that the outcome of the attack would be worth the attacker's time invested, subsequently increasing the probability of an attack itself. Therefore, it is critical for both users and organizations to ensure that they have the proper security protocols in place and that they understand how to use those security protocols if the situation arises. This study will focus on two such security protocols, why they are important, and how they can be done as effectively as possible.

Wiping your device

As discussed, there are many vectors that an attacker can take to maliciously attack an individual or organization. While it is improbable to fully mitigate risk in an organization there are a few steps that individuals can take to help guard against a successful attack. One of those steps is to ensure that a device can be 'wiped' if lost. Wiping a phone entails either removing all personal data (effectively restoring the device to its original state when first purchased) or removing all data off the device entirely. The latter option typically makes the device's data unreadable and the device itself unusable.

To complete a successful wipe a user must ensure that a program with remote wiping capabilities is present on the mobile device. Apple devices come standard with 'MobileMe' which can be used to locate lost devices and remotely wipe them if necessary. There are many third party applications available for the Android platform that allows the user to complete the same requests. This study looked

at one such program, PhoneLocator Pro, to verify its ability to successfully wipe the device. To wipe the phone, the user simply needs to text the “wipe passcode” to the device from any phone number. The application’s default wipe code is PLwipePL, shown in Figure 46, which can be customized for security. Once the phone receives this code it immediately shuts down and begins to wipe user data from the phone. It can also be programmed to wipe data from the external SD cards connected to the device. After testing the application on the study’s HTC Android, the application was proven to work as described. Of the confidential data that was discovered during the initial forensic examinations, none of the passwords, photos, email addresses, contacts, or SMS messages were available after wiping the device, nor was any other data available after the wipe was completed.

While wiping is a great way to protect your device, it must be done before the attacker has the opportunity to image the device. If it happens after the attacker has imaged the device, the attacker will have already gained access to the user data stored on the phone and the wipe will be virtually meaningless. Also, if the attacker is able to gain access to the device and disable the network connections (i.e. 3G and 4G, Wi-Fi, etc.) then the device will be unable to receive the wipe code, rending the device helpless in the hands of the attacker.

After an Android device is wiped, the owner has the option of syncing back to Google’s servers to obtain stored data such as contacts, calendar information, and more. If the attacker has access to this account information, the user must change the password to the Gmail account tied to the phone. Otherwise, the attacker will be able to sync the phone with the original Gmail account and download the victim’s data backed up with Google back to the device.

There are also companies that specialize in device wiping and reselling. Gazelle.com, a tech reseller based in Boston, specializes in used mobile device sales. According to Kristina Kennedy, a manager from Gazelle.com, “’50 to 65 percent of phones that come to Gazelle’s warehouse each day have the previous owner’s information in them.’ To deal with that, the company trains its staff to perform a manual factory reset on each device that comes through the door, along with destroying any SIM cards and formatting SD cards that may arrive with the devices.” [25] PC World purchased phones from various companies that claim to wipe phones before reselling them, testing the company’s effectiveness. They purchased a phone from Gazelle and found it completely clean of information. This is in stark contrast to the company G0g0gadgets, from whom PC World bought an LG Dare and found it loaded with personal text messages, pictures, and other data leftover from the original owner. As with any situation, precaution should be first and foremost. The user should wipe the phone clean and ensure that it is clean themselves before sending the phone to a third party for reselling. [25]

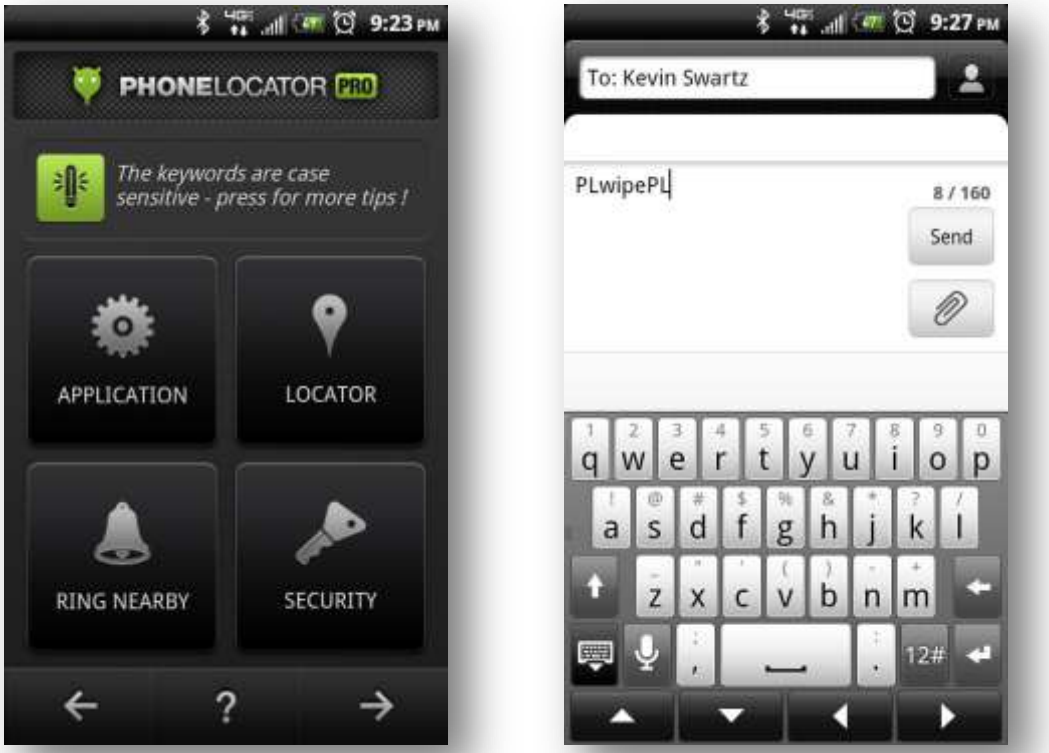

Figure 46: Left: PhoneLocator Pro application screen shot. Right: text message initiating phone wipe)

Update, Update, Update

Apple and Google are routinely pushing updates for their devices. Among the reasons the manufacturers consistently push these updates include product enhancements as well as device and network security. Apple and Google (as well as all other mobile OS developers) are routinely testing their devices for security flaws and patch those flaws with updates at regular intervals of time, either through an over the air (OTA) update or by requiring the user to download new firmware through a computer based application (iTunes, for example). These updates both enhance the user experience (adding new features, streamlining the visual experience), and plug any newly discovered security holes in the OS. A device that is out of date from the current OS runs the risk of having a security flaw remain on the device that can be exploited by an attacker.

As an example, a serious threat nicknamed “Android.Bmaster” and “GingerBreak” was discovered on Android Gingerbread devices running versions 2.3.4 and before as well as Android devices running version 3.0. According to Mathew Schwartz of Information Week, as of February 2012 the malicious application appeared to be on approximately 11,000 Android devices. It was initially extremely effective at circumventing security controls while also being extremely difficult to discover by users, anti-virus software, and OS security features alike. This is because it names itself “com.google.android.smart” in the Android OS’s file structure. That is the exact same name as a settings app used by the native Android operating systems. Once on the device the application attempts to gain root access. Whether it is successful in gaining root access or not, the application then remotely connects to a server which attempts to push through malicious applications to the device. These

applications can be used to send SMS messages to “a particular premium SMS number at a specific rate (three a day, for instance) for a certain number of days”, generating revenue for the attacker. [26] Google stated that “since May 2011, all Android device updates have included a patch against GingerBreak”. [26] The issue that has arisen, however, is that carriers are required to push the Android updates remotely to their subscribers. While Apple has been consistent in their device updates, Android mobile carriers have not. [27] This can create security vulnerabilities discussed earlier, enhancing the need for other security settings on a device outside of exclusively relying on regular updates from the manufacturer or mobile phone carrier.

The Android Software Developer’s Kit (SDK)

The Android Software Developer’s Kit can be an invaluable tool for a forensic examiner. It was created to allow software developers unfettered access to Android devices to create, implement, test, and enhance applications built for the Android platform. Both the Android Virtual Device Manager and Android Device Emulator are applications downloaded with the SDK that allow the user to emulate any Android OS that has been created and released to the Android community as well as all OS’s that have been rolled out to the market. Since there are many variations of Android operating systems on the market, the SDK provides a forensics expert with access to every Android OS without having to physically acquire an individual device for each version and each OS. Forensics experts can then use the emulator to add software and data with the purpose of eventually parsing through the file structure to see where data is stored.

While the SDK is an incredibly powerful tool, the downside to the SDK is its limited Android Market capabilities. While it does allow importation of SDK versions of popular market apps it, does not allow a user to download the native market versions of popular applications such as Facebook, Twitter, and others through traditional means. It was for this reason that the SDK was not utilized in this study. It is, however, an incredibly powerful tool, one that can be utilized to study each Android OS in depth. For that reason it is highly recommended to be part of any forensic examiner’s toolkit. Screenshots of the SDK running in Ubuntu are provided below (see Figure 47, Figure 48, and Figure 49) for reference.

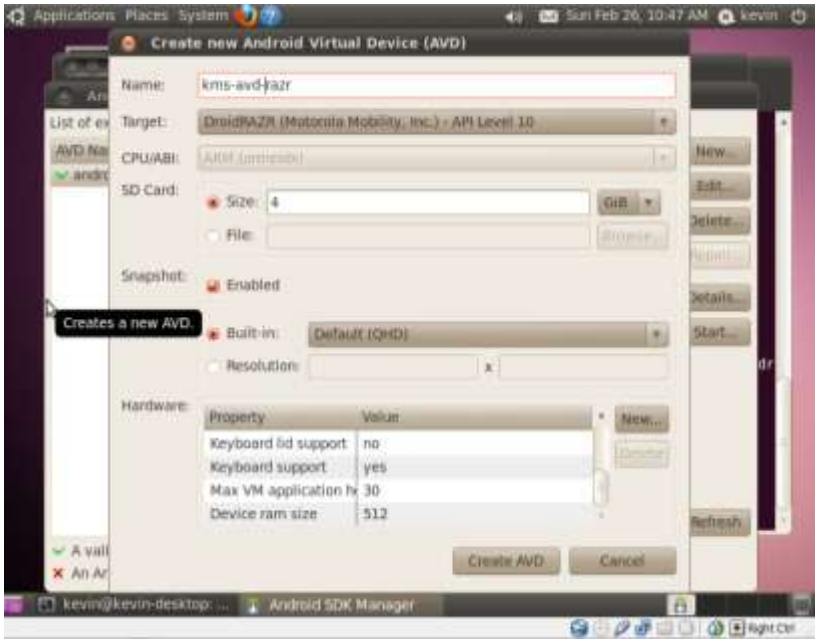

Figure 47: Creating a new Android Virtual Device

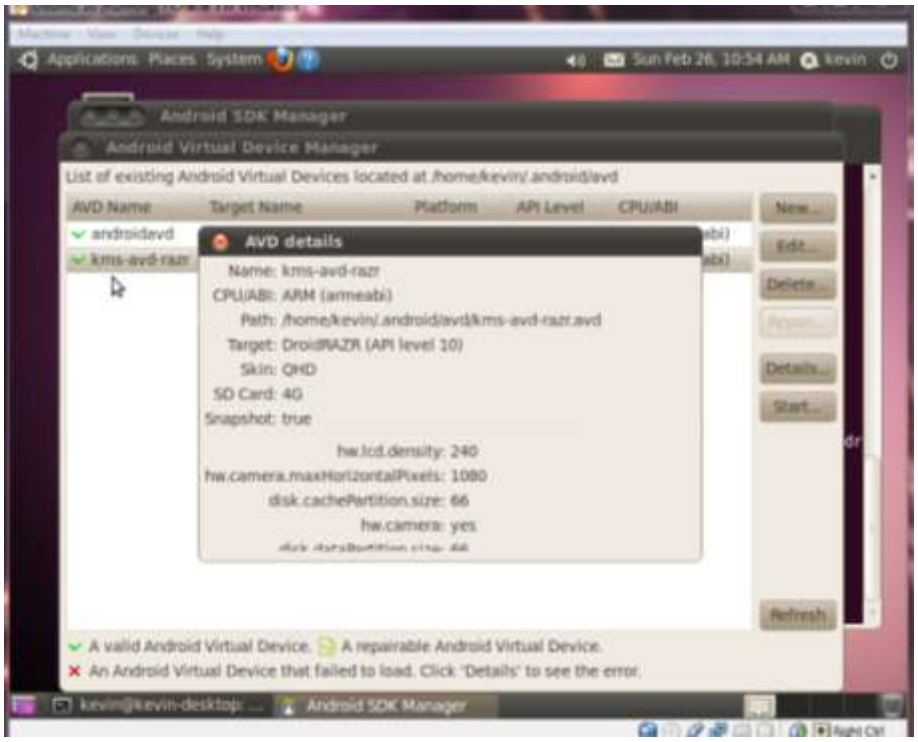

Figure 48: Creating a Droid RAZR in the Android SDK's Virtual Device Manager

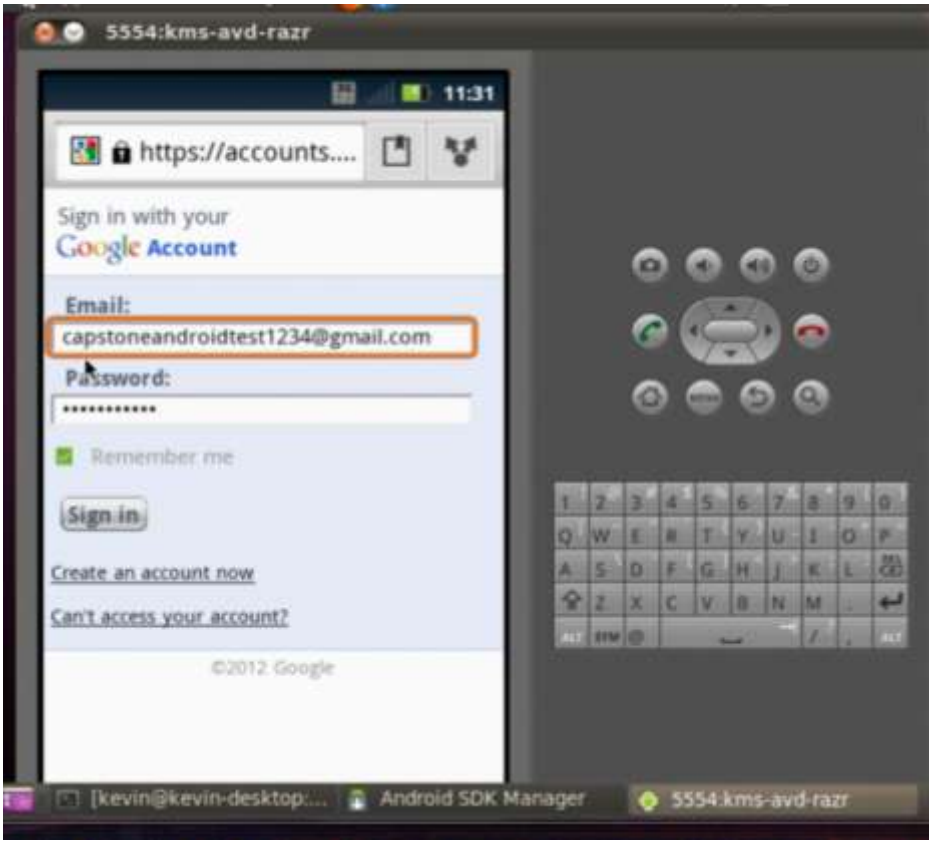

Figure 49: The Device Emulator in the Android Software Developer's Kit.

Conclusion

While the future of mobile devices is as uncertain as the future of technological advances themselves, one thing is certain: as long as confidential data exists there will be attackers trying to obtain it. Mobile device use is erupting and, combined with the subsequent explosion of social media websites and their mobile applications, pose new, never-before-seen threats to individuals and businesses alike. Personal and confidential data is posted online, sent over networks, and stored on devices. Most users are not aware of the variety of risks that are present with keeping a large amount of personal data on a mobile device until the moment their data is lost or stolen. Typically, if the user's device is lost the opportunity to mitigate risk or prevent an attack has already passed. It is for this reason that users need to be educated, devices need to be consistently updated, and preventative measures need to be taken. Also, phone manufacturers, wireless carriers, and application developers need to ensure that security is a central focus of the device, OS, or application that they are creating. While the world of technology will continue to advance at an extremely rapid pace, users and companies need to stay on the forefront of safety and security. If they do not, they may find themselves on the wrong side of an information leak that cripples their most confidential data or their organization's data integrity as a whole.

Bibliography

[1] L.A. Times. (2011, December) latimes.com. [Online]. http://latimesblogs.latimes.com/technology/2011/12/google-facebook-youtube-are-2011-most-visited-websites.html

[2] David Kirkpatrick, *The Facebook effect: the inside story of the company that is connecting the world*. New York, United States of America: Simon & Schuster Paperbacks, 2010.

[3] Mary Smith and Chris. Treadaway, *Facebook Marketing: An hour a day*, Willem Knibbe, Ed. Indianapolis, United States of America: Wiley Publishing, Inc, 2010.

[4] Nielsen Wire. (2012, January) nielsen.com. [Online]. http://blog.nielsen.com/nielsenwire/consumer/more-us-consumers-choosing-smartphones-as-apple-closes-the-gap-on-android/

[5] Roberta Bragg, *CISSP: Certified information systems security professional*, Jeff Riley, Ed. United States of America: Que Publishing, 2003.

[6] Andrew Hoog, *Android Forensics: Investigation, Analysis, and Mobile Security for Google Android*, Angelina Ward, Ed. Waltham, United States of America: Syngress, 2011.

[7] Andrew Hoog and Katie Strzempka, *iPhone and iOS Forensics: Investigation, Analysis and Mobile Security for Apple iPhone, iPad and iOS Devices*, Angelina Ward, Ed. Waltham, United States of America: Syngress, 2011.

[8] Sean Morrissey, *iOS Forensic Analysis*, Michelle Lowman, Ed. New York, United States of America: Apress, 2010.

[9] Adrian T.N. Palmer, "Computer Forensics: The Six Steps," Kroll Ontrack Computer Forensics, Newsletter.

[10] Eric Zeman. (2012, February) Android's Success: By The Numbers. Online Article. [Online]. http://www.informationweek.com/news/mobility/smart_phones/232601613

[11] Cyber Warrior. (2012, January) Android Central. [Online]. http://forums.androidcentral.com/thunderbolt-rooting-roms-hacks/123060-guide-how-root-unroot-thunderbolt-revolutionary.html

[12] AccessData. (2012, March) MPE+: Mobile Forensics. [Online]. http://accessdata.com/products/computer-forensics/mobile-phone-examiner

[13] Access Data Group LLC. (2011) FTK User Guide. PDF Document.

[14] Steven Bolt, *Xbox 360 Forensics: A Digital Forensics Guide to Examining Artifacts*, Angelina Ward and Heather Scherer, Eds. Burlington, United States of America: Syngress, 2011.

[15] Federal Communications Commission. (2012, March) Federal Communications Commission: Audio Division. [Online]. http://transition.fcc.gov/mb/audio/bickel/DDDMMSS-decimal.html

[16] Peter White, *Crime Scene to Court: The Essentials of Forensic Science*. Cambridge, United Kingdom: The Royal Society of Chemistry, 2010.

[17] Inc. Apple. (2011, October) Apple.com. [Online]. images.apple.com/ipad/business/docs/iOS_Security.pdf

[18] Access Data. (2012, February) Access Data. [Online]. http://accessdata.com/downloads/media/MPE_FAQ_v7.pdf

[19] Michael E. Whitman and Herbert J. Mattord, *Principles of Information Security*, 3rd ed., Steve Helba, Ed. Boston, United States of America: Course Technology, 2009.

[20] Elinor Abreu. (2000, September) Kevin Mitnick Bares All. Online Journal. [Online]. www.nwfusion.com/news/2000/0928mitnick.html

[21] Aaron Marcus, *Designer, User Experience, and Usability: Theory, Methods, Tools and Practice*. New York, United States of America: Springe, 2011.

[22] Ted E and viaForensics. (2011, December) viaforensics.com. [Online]. https://viaforensics.com/mobile-security/question-does-ios-encryption-work-protect-data.html

[23] MarketWatch. (2012, April) MarketWatch.com. [Online]. http://www.marketwatch.com/story/personal-information-is-top-target-of-cyber-attacks-new-cdw-research-shows-2012-04-17

[24] Pamela Lewis Dolan. (2012, April) AMedNews.com. [Online]. http://www.ama-assn.org/amednews/2012/04/02/bisf0405.htm

[25] Megan Geuss. (2011, July) PC World. [Online]. http://www.pcworld.com/article/235276/your_old_smartphones_data_can_come_back_to_haunt_you.html

[26] Matthew J. Schwartz. (2012, February) Information Week. [Online]. http://www.informationweek.com/news/security/mobile/232600576

[27] Matthew Schwartz. (2011, November) Information Week. [Online]. http://www.informationweek.com/news/security/mobile/232200144

[28] Google. (2012, January) source.android.com. [Online]. source.android.com

[29] SANS. (2012, January) Sans.org. [Online]. computer-forensics.sans.org/community/downloads

[30] Larry Daniel and Lars Daniel, *Digital Forensics for Legal Professionals*, Chris Katsaropolus, Ed. Waltham, United States of America: Syngress, 2012.

[31] Frank H.P. Fitzek and Frank Reichert, *Mobile Phone Programming and its Application to Wireless Networking*. Dordecht, The Netherlands: Springer, 2007.

[32] Marie-Helen Maras, *Computer Forensics: Cybercriminals, Laws, and Evidence*. Sudbury, United States of America: Jones & Bartlett Learning, 2012.

[33] The Association of Chief Police Offiers. (2012, February) Good Practice Guide for Computer-Based Electronic Evidence. [Online]. http://www.7safe.com/electronic_evidence/ACPO_guidelines_computer_evidence.pdf

[34] James Sammons, *The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics*, Chris Katsaropolus, Ed. Waltham, United States of America: Syngress, 2012.

[35] Michael Sheetz, *Computer Forensics: An Essential Guide for Accountants, Lawyers*. Hoboken, United States of America: John Wiley & Sons, 2007.

[36] John R. Vacca, *Computer Forensics: Computer Crime Scene Investigation*, David Pallai, Ed. Hingham, United States of America: Charles River Media, 2005.

[37] John R Vacca and K Rudolph, *System Forensics, Investigation, and Response*, Lawrence J Goodrich, Ed. Sudbury, United States of America: World Headquarters, 2011.

[38] Jonathan Zdziarski, *iPhone Forensics: Recovering Evidence, Personal Data & Corporate Assets*, Andy Oram, Ed. Sebastopol, United States: O'Reilly Media, Inc., 2008.