2012

First Responder Assistance Tool for Mobile Device Forensics

Benjamin Kallal
Iowa State University

First responder assistance tool for mobile device forensics

By

Benjamin Kallal

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-majors: Computer Engineering; Information Assurance

Program of Study Committee:
Yong Guan, Co-major Professor
Doug Jacobson, Co-major Professor
Suraj Kothari

Iowa State University

Ames, Iowa

2012

Contents

List of figures

Abstract

The growing importance of mobile telephones, especially so called "smartphones", the problem of identifying these phones has become a real issue. There is a prevalence of these devices being used by criminals, foreign agents and terrorists. The need to be able to quickly identify these phones and determine what forensics tools maybe compatible with the device is critical. The issue of imitation phones and the potential of hidden operating systems have further muddied the forensics waters. Having a starting point from which to perform this analysis is important first step. The purpose of this research is to provide that starting point. By analyzing basic aspects of the phone and viewing the compatibility with other forensics tools it will give the investigator ideas as to what they can reasonably expect to gain from analyzing the phone. Additionally this software attempts to gather information about the software with the intention of detecting hidden partitions and secondary operating systems.

Chapter 1: Introduction

As smartphones become more and more common, the need to gather data from them for investigative purposes has become critical. Imagine a situation where a young girl has been kidnapped from a rural country home. The kidnapper has made a mistake though; he accidently left his phone behind. How important would it be for the first responders to quickly be able to identify the phone and determine what they could do to get information before it is too late?

One of the major issues facing forensic examiners and first responders is to know what device exactly they are dealing. With the number of devices on the market growing almost exponentially knowing what exactly they are dealing with is becoming a difficult task. Another major issue is being aware of what sort of techniques are compatible with that device. With the cost of forensic software avoiding purchasing the wrong equipment is extremely important.

The first responders and forensic examiners could be greatly aided by this software, which provides the ability to identify a device using a fingerprint and also provide information about compatible forensics tools and techniques.

This proposal involves using the idea of device fingerprinting to identify a singular device as well as its model. Using that fingerprint we link to a database containing pertinent forensic tools and techniques. This allows for quick identification of the device and also knowledge of the techniques that can be applied to the device to gain valuable data. We tested our prototype by using a selection of Android devices to test the feasibility of the fingerprinting process.

As we show later we were able to successfully gather the necessary information to create the fingerprint of a device. The forensic examiner using this software could potentially save time and effort, increasing a lab’s level of productivity.

Chapter 2: Problem Definition

Aside from the situation proposed above, the number of phones available on the market has resulted in a multitude of software packages that offer the ability to analyze and forensically extract data for use in investigations. These packages can often cost thousands of dollars per license and thousands more in overall training costs for each examiner in the lab. This cost puts these extremely important and useful software packages outside the financial reach of many small police and IT security departments. Therefore, a readily available database and desktop client will provide police departments with a starting point to determine if they should handle the forensic analysis themselves or hand it off to another larger lab, such as the state or federal government forensics labs. This could save agencies both time and money and in the current budget climate, both of those resources are at a premium.

Further complicating the situation mobile phone's hardware and software can now have different operating systems running on the same device. For instance, there are several known methods for getting Android to work on a Windows Mobile device. Obviously, this has an effect on most forensics tools and could easily cause an issue where normal forensics tools could accidentally render the device inoperable. Having a fingerprint could prevent these problems. Secondary operating systems can easily be used to hide criminal activity from a basic forensics probe since current tools do not provide such an ability. The ability to look at a phone's memory and what is available would provide a useful feature to law enforcement when analyzing a device for evidence. Additionally, this opens a new field for exploration which has not been addressed much outside of the phone hacking community, at sites such as XDA-developers [1]. By being able to analyze devices, law enforcement will be able to detect activities by more technically minded individuals. For example, if the police confiscated a Touch Pro 2, which normally runs Windows Mobile, it could also potentially be running Android [2]. A poorly trained forensics investigator might just turn on the phone, see that it is running Android, and never understand that there is a Windows Mobile OS and data on the phone. This could result in the loss of important forensics data and break the case against the defendant. The fingerprinting software could help an examiner quickly determine if there is a chance the device is running a different operating system, because it will not match the fingerprint for that particular model and give the resulting compatible forensics techniques.

The number of forensics tools on the market today for use by anyone from law enforcement to private investigators has ballooned with the now ubiquitous smartphone. With smartphones now approaching almost 50% of the mobile phone market, these advanced devices require advanced tools. Indeed, today's smartphone is undeniably more of a computer than a phone. The tools used to analyze them are produced from companies that range in size and authority. They range from that of Encase, the makers Guidance Software, to smaller companies who offer boutique packages with a specific scope of operations such as iXAM, for the iPhone makers, Forensics Telecommunications Services Ltd. These tools often involve a great cost, strict licensing agreements, and a complex interface which can require hundreds of hours of training to become proficient in.

This leads to a situation where many agencies, especially small agencies, are unable to support the full spectrum of mobile device forensic tools. Take the story at the beginning, for example. If the kidnapping happened in a rural county, the law enforcement agency would very likely not have the tools needed to analyze the recovered device. Additionally, the agency would have little experience working with these devices. If they were to use this software then they would be able to determine what options they had available to successfully extract the data from the system. From there they could decide if it was worth trying to access the device themselves or if they should hand it off to a different organization.

In summary, there are several key problems facing the mobile device forensics field:

- There are thousands of devices currently on the market.
- There are many operating systems and a phone could be running an unexpected one, or multiple ones.
- There are many tools for many situations which may or may not be adequate.

- The explosion of smartphones has outstripped the ability of many departments to keep up with the growth, either for financial or educational reasons.

In order to prove useful to a first responder a tool must meet several distinct requirements:

- Accurately identify a phone, both the model and a specific universally identifiable.
- Require a minimum of information to avoid triggering issues with security features or accidently damaging the device.
- The ability to provide information about compatible forensics tools and techniques for a given device.

Use Case

There are two main use cases that are available to the end user. The first would be a first responder who would want quick information as to what they could conceivably do without specialized equipment. Alternatively, in a more controlled setting the forensics examiner would utilize the tool as a starting point.

The first responder would want to use this software to quickly analyze a phone if a forensic examiner is not readily available or the phone battery may not last long enough. This software would give the first responder a quick way to check and see if there is any way to reasonably gather data off of the device without waiting for a regular forensic examiner. The advantage would be that the user would be able to quickly identify the phone and recover evidence in highly time sensitive situations, such as kidnappings.

The second user for this would be a standard forensic examiner. This type of user would primarily be interested in using the software for a quick reference guide and starting point for a digital forensics examination. They would see what kind of options they have available for examination. Also, they could see if the fingerprint matches what they the expected, giving an indication that the phone is a knock off and standard techniques may not work.

Chapter 3: Background

Mobile Device Forensics

The field of mobile device forensics has developed rapidly in the past five to ten years as cellular devices have rapidly become the backbone of many people's personal life. These devices now frequently contain a high degree of detail about a person's life. Such as call history, text messages, emails, contacts, tasks, calendar entries and on newer phones even location history letting forensic examiners gain lots of information about a suspect by simply analyzing the device. This can save valuable time and money when done correctly by a fully trained forensic examiner. This has created a huge field of forensic software being developed by a wide range of forensic software companies.

Typically speaking there have been several levels of forensic analysis and within that different types of analysis. One definition of these levels was put forth by Sam Brothers [13]. They are: manual extraction, logical analysis, physical extraction, chip-off and micro-read. Manual extraction is to actually use the phone by hand and record the information acquired by video [13]. Logical analyses is using some form of connection between the device and computer then use various protocols to acquire the desired knowledge [13]. Physical extraction is where the entire memory of a device is extracted rather than just the actively used memory [13]. The challenge at this level is that the data is not stored in an easily readable format. Chip-off involves physically removing the memory chips and then extract the data. The drawback here is similar to that of physical extraction [13]. The final level micro-read involves actually taking the chip and reading the exact values of the gates, this level has issues with being extremely difficult and time consuming [13]. Typically one does this sort of analysis when the device have been physically damaged.

Additionally the National Institutes of Standards and Technology has published information about what they consider important for the use of mobile device forensics. Their “Guidelines on Cell Phone Forensics” was published in May 2007 [9]. Also they published “Cell Phone Forensic Tools: An Overview and Analysis” in October 2005. These two documents provide the basis of the federal government’s opinion for mobile device forensics.

The Guidelines document discusses the importance of how organizations need to organize themselves for the development of a clear policy for considerations and procedures on mobile devices. The next two major points they make are clear points that are that organizations should have reasonable support for mobile devices and professionals who are trained to do such forensic examinations[9]. The idea of reasonable support in the mid-2000’s is a very different aspect than “reasonable” in 2011, it most likely would be almost an order of magnitude more than before.

Guidelines goes into detail about what sorts of information is recovered off a device. The core ideas behind this is the process of doing “physical acquisition" and "logical acquisition". These are similar to the previously discussed logical analysis, and physical extraction from Mr. Brothers.

Physical acquisition is a bit-by-bit copy of the device's memory. Physical acquisition allows deleted files to be accessed as well as all un-allocated space. This creates an image file that can later be analyzed and carved by a forensic tool. Ideally a physical acquisition is what is achieved however the number of tools that are able to acquire a physical acquisition is less than that can achieve a logical acquisition [9].

A logical acquisition is more limited but it is easier for a tool to be developed to acquire that information. It also provides a more a better organized result because it is acquiring that which is being used currently.

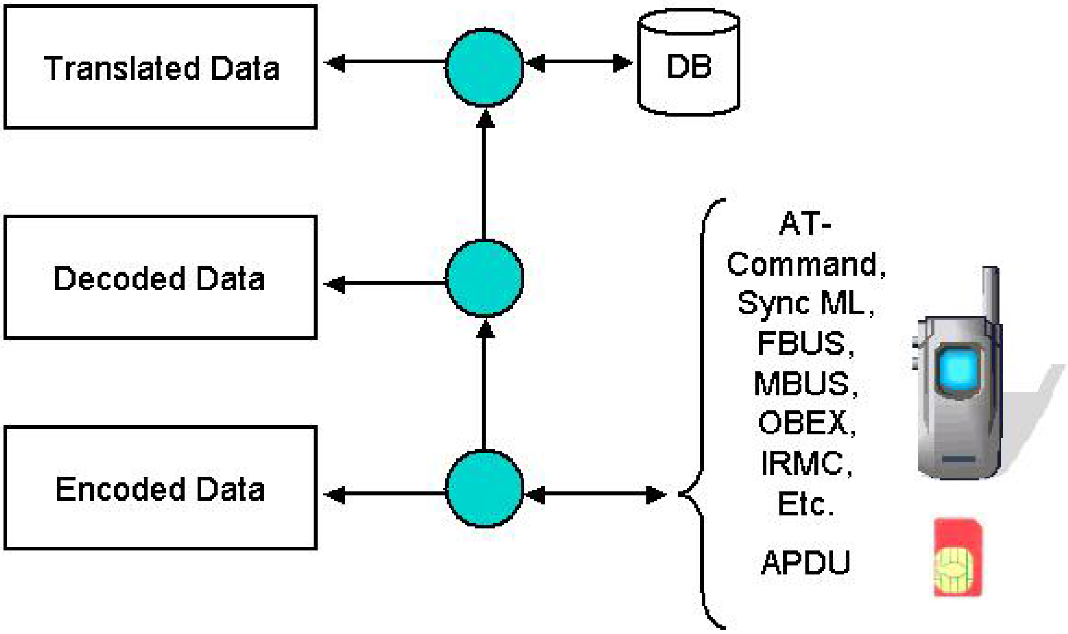

Figure 3-1

High level diagram of forensic analysis source [9, 13]

As can be seen be seen in the figure above HyperTerminal commands are commonly used for some of the acquisition. These commands were some of the first techniques used for the development of forensic analysis.

The “Guidelines on Cell Phone Forensics” further goes onto discuss several key principals for acquiring electronic evidence from the “Proposed Standards for the Exchange of Digital Evidence” [9, 25]:

- Upon seizing digital evidence, actions taken should not change that evidence.
- When it is necessary for a person to access original digital evidence, that person must be forensically competent.
- All activity relating to the seizure, access, storage, or transfer of digital evidence must be fully documented, preserved, and available for review.
- An individual is responsible for all actions taken with respect to digital evidence while the digital evidence is in their possession.
- Any agency that is responsible for seizing, accessing, storing, or transferring digital evidence is responsible for compliance with these principles.

Another important not that the Guide discusses is from the Electronic Crime Scene Investigation – A Guide for First Responders created by the Department of Justice [9, 26]:

- Securing and Evaluating the Scene – Steps should be taken to ensure the safety of individuals and to identify and protect the integrity of potential evidence.
- Documenting the Scene – Create a permanent record of the scene, accurately recording both digital-related and conventional evidence.
- Evidence Collection – Collect traditional and digital evidence in a manner that preserves their evidentiary value.
- Packaging, Transportation, and Storage – Take adequate precautions when packaging, transporting, and storing evidence, maintaining chain of custody.

The areas of securing and evaluating and also the documenting the scene are the basis of where this research is based on. Helping improve that identification is key part in successfully completing a forensic examiner's work.

Examples of Mobile Device Forensic Software

Some of the most well-known and popular software products are: Paraben, Celdeck, Secure View, and xry. On top of this there are a pleather of smaller device manufacturers such as the smaller iXam.

Figure 3-2

Paraben's Device Seizure [10]

Paraben makes a product called “Cell Seizure” which works on various phone types, such as GSM and CDMA. The extracted data is sotred in a proprietary format, making using other software more difficult. The kit comes with equipment to contact numerous types of phones. Cell Seizure supports both logical and physical acquisition [9, 20]. It costs $1,795 per license and 360 dollars a year in subscription fees. [10]

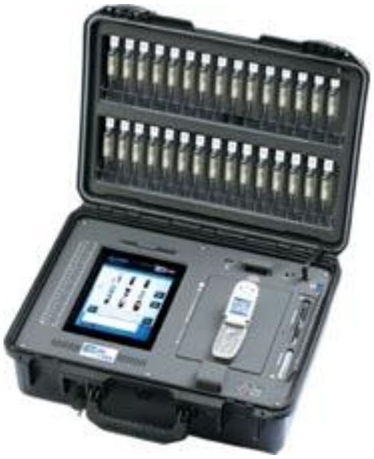

Figure 3-3

Cell DEK Field Kit [15]

Cell DEK is made by Logicube and comes with a rugged carrying case and a built-in touch screen computer. The system also contains a large set of cables to access the device. It allows the reports to be created in a custom formatted HTML file with recovered data [9, 20]. Cell DEK costs $12,000 dollars for the base model but does not have a subscription fee [10].

Figure 3-4

Secure View Field Kit [16]

Secure View is made by a company called Susteen and was originally based on a phone management software. It only supports logical extraction. The software does not secure extracted data beyond a password. It also allows the searching of acquired data for easier analysis [9, 21]. For a single license costs $2,499 with a cost of $999 for a yearly renewal fee [10].

Figure 3-5

.XRY Field Kit [18]

Micro Systemation makes a suite call .XRY. It extracts data from both GSM and CDMA devices. It requires a USB dongle to allow operation, to help reduce piracy. Like the other software suites it contains the cables necessary to connect the devices to a computer. The software uses hashes and passwords to maintain records of how the case data was used [9, 21]. .XRY is offerend in two versions one is logical and the other is physical. For logical it costs $3,950 initially for $1,990 per year [10]. For the physical acquisition model it costs $5,950 and $2,900 dollars a year in subscription fees [10].

Device Fingerprinting on Other Device Types

There has been a lot of work done in the field of device fingerprinting. However much of this work has been concentrated around the idea of fingerprinting a device using a wireless card. There has been some work done on fingerprinting a device using a physical cable. One such example of this is the paper “Host Identification via USB Fingerprinting” [14].

The author’s proposal assumes that USB stack provides enough information to identify a specific computer because of the information that is available when connecting the device. It states that each layer uses enough differences that you can determine different devices based on the way the host side interacts with a USB device.

The first part of their solution includes USB analyzer. They used the Ellisys USB Explorer 200 which forwards the communications to a software program which records the USB traffic. When they tested their idea they used campus computers to identify things. They analyzed Windows, OS X and

Ubuntu. They used several different types of USB devices to such as a mouse and a USB stick. They ran their experiment using the following steps:

1) Make sure the computer is switched on and displaying the login screen. For this work, we did not analyze the possible effects of time-since-bootup.

2) Disconnect all USB devices to eliminate cross device interference.

3) Connect the host computer to the USB analyzer, using the same port within machine types.

4) Record traffic data for 15 seconds.

5) Disconnect the device and save the USB trace.

6) Perform 15 fingerprinting trials for each machine.

[14, 3]

They figured out that they were most interested in two type of USB communication: IN/OUT transfers and enumeration. Enumeration is the initial handshake when a USB device is attached, this allows the software to see the differences in how USB devices are discovered. The IN/OUT transfers occur after the device configuration. This is meant to be used in the future to increase granularity of the suite.

Additionally there are several steps they take to identify the hardware as well as the operating system. They use timing data to try to determine that model. Namely they use: Suspend, reset, and retry.

After running their test on numerous devices they show that using a USB interface is a reasonable way to identify a computer.

Related Work in Mobile Device Forensics Fingerprinting

There are several products on the market that provide some similar features; however, they have some downsides. The most closely related product is Cellebrite’s UFED Physical Analyzer. However, this has a problem of it relies on the visual accuracy of the user. This lends itself to having issues with the idea of counterfeit phones. While the fakes look extremely similar they are not legitimate. This could lead to either a damaged device or a waste of time. Additionally it leaves off the issue of providing a comprehensive compatibility database and largely locks you into Cellebrite's products. Finally, it does not provide the ability to identify a single particular device. [7]

Another of Cellebrite’s products is their UFED Physical Pro. This device is also designed to help ease the process of a forensic examiner, however, it seems to have some similar drawbacks of not being as supporting of the idea of custom operating systems. [8]

XRY Local is another piece of software that provides some identification of a device based on some pre-defined ability to determine what kind of phone it is, but again it would not have any support for dealing with a phone that is not running an expected operating system and has predefined settings for iPhone and Blackberry devices, a mistake in selection there could cause issues with the physical device.

Chapter 4: FFAAD (First-Responder Fingerprinting and Database)

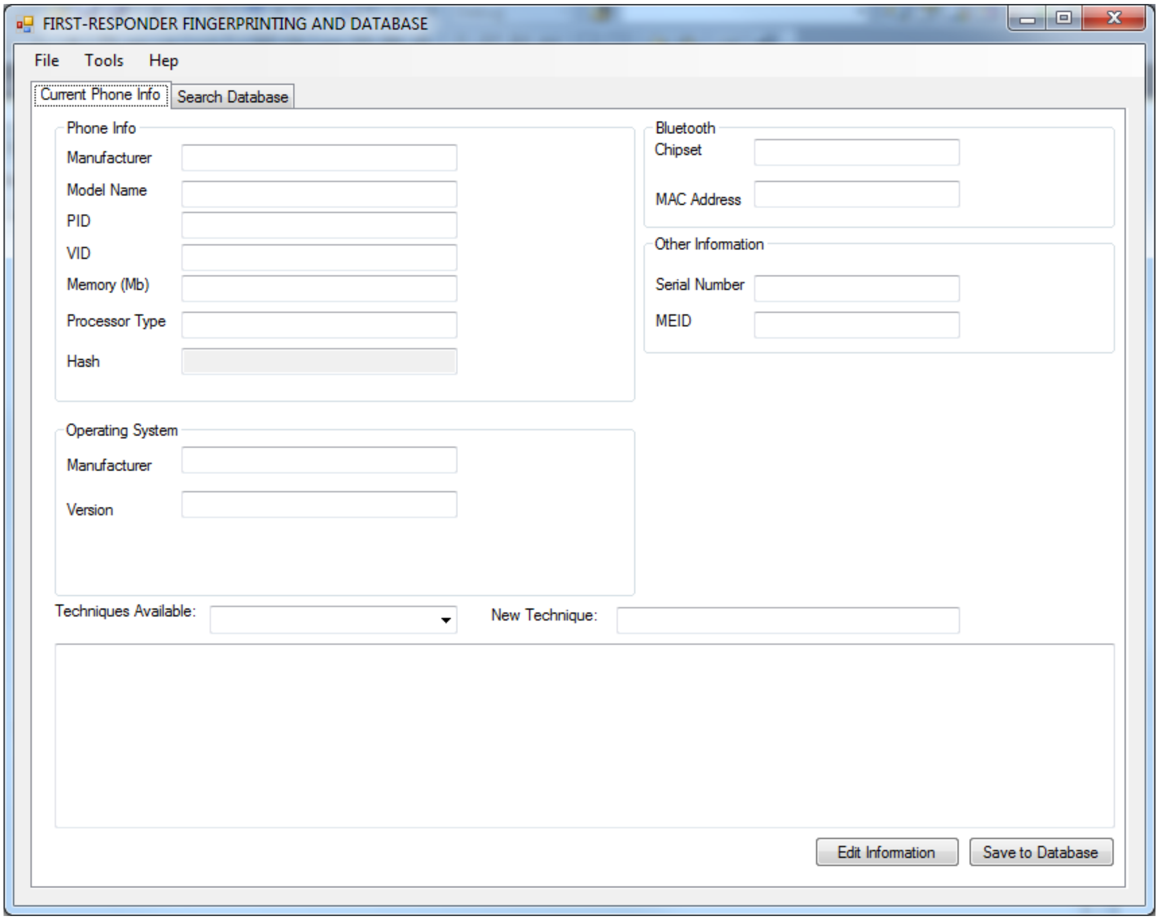

Figure 4-1

Main GUI of the FFAD Software Suite

The primary objective that this software suite is trying to show is that is possible to achieve identify an individual phone using simple commands so as to not accidently damage the device. From there, the goal was to use that identification to link to the software compatibility database and determine what software works best with a given device, while attempting to achieve a minimum level of interaction. Currently, there is a not a software package available that combines the ability to fingerprint a device or support the ability to use that fingerprint to determine vetted digital forensics tools and techniques.

We define a “fingerprint” as being an electronic signature that can be used to identify a single phone from all other known devices. The fingerprint is also meant to use the minimum amount of data that cannot reasonably be changed by the user and still actually produce a specific fingerprint.

We address the requirements by using a multi-part program. We address the first two requirements with the first two sections of the solution; the first deals with USB interactions and the other handles operating system specific tools. These two areas handle the first two requirements. The last requirement is met by the use of a database to identify the tools and techniques that are known to work with a particular device, handling the third requirement.

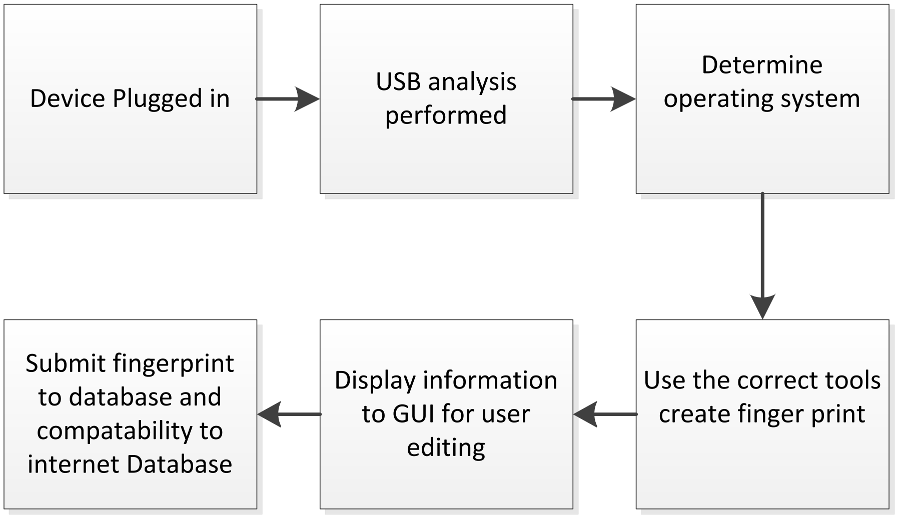
Device Plugged in
USB analysis performed
Determine operating system
Use the correct tools create finger print
Display information to GUI for user editing
Submit fingerprint to database and compatability to internet Database

Figure 4-2

Flow diagram of process of fingerprinting

The fingerprinting process

This section discusses the area of doing actual fingerprinting on the phone device itself. However, due to time limitations the only thing that was fully implemented was on the Android operating system.

Importance of fingerprinting

There are many additional features that this software can be used a base to expand into. The concept of fingerprinting could be expanded out to do things such as create a hash of an entire operating system or a particular part allowing an investigator to quickly note any area specific changes a user may make.

From the law enforcement perspective, one could make a fingerprint of a phone discreetly from a criminal. After the fingerprint has made they can then reinsert the phone into the wild. After that, when an arrest is made, the forensic examiners can quickly detect changes and then will have specific areas to look at. Again, the advantage of doing simpler fingerprinting is that the relatively low level of interactions necessary to do this will make it less likely to make a change the suspect might notice or trigger a security feature.

Moving forward the ability to ensure that a phone is what it appears to be and is running the software that it is advertised as running will become increasingly important. By using and expanding on this platform investigators and individual users will benefit in a way that has not been previously offered by known current solutions.

From an end user's personal or corporate security the application could serve a similar purpose in that would let them know if their phone had been somehow compromised. The most obvious example would be an employee who has sensitive data on their mobile device and the employee needs to travel overseas. Prior to going overseas the phone could be fingerprinted and recorded. This would provide a baseline for the device and its status. The employee would then take their device overseas and upon return, the device would be re-fingerprinted. If the fingerprints do not match then the IT security staff would know immediately the device had been compromised. They could then proceed to take steps to mitigate any introduced risks.

So called “white box” phones and KIRF (keeping it real fake) phone as termed by Engadget [4] also present a growing challenge to law enforcement. Essentially, these are knock-off products from third world countries such as China and India. These fake devices are often designed to look exactly like more well-known phones such as iPhones and Samsung Galaxy phones. These phones are well known in places such as the United States and would be easily recognized by many forensic examiners. However because they might be using different versions of either iOS or Android, respectively though could actually damage the device or remove the ability to be used in court trying to perform the normal forensics procedures.

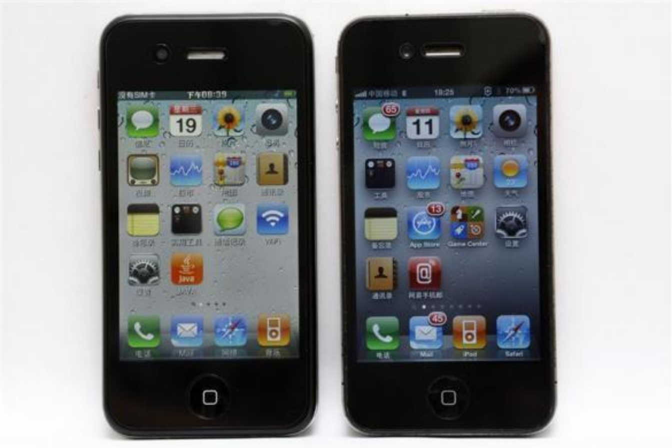

Figure 4-3

An example of how close a fake and real iPhone can be, note how the even the icons are very similar. [17]

USB Interfacing

The USB standard has several different aspects which may or may not be implemented. These include things such as a data standard, and also a power standard. Because the goal is to minimize the dependencies for the fingerprint, only some of the optional aspects are considered. Some of the included information included in the standard is a product and vendor identification number. The vendor identification tells the maker of the device. In the case, the mobile devices that where reviewed in this project, show the actual manufacturer of the device. The product ID is then directly related to actual mobile device. This provides a good starting point for fingerprinting.

The first step that is taken when plugging in the suspect device into a machine is to use the LibUSBDotNet USB wrapper for .Net to get information such as the product and vendor IDs[3]. Using this library it is possible to detect when an USB device is plugged into a machine. Then the product ids and vendor ids can quickly be found. From this, determining what operating system the phone is running, and can be reasonably guessed. For example, if the vendor ID is from Apple, it almost certainly runs iOS, or Motorola it almost certainly runs Android.

Once a suspected operating system has been identified, the application will then route it to the proper operating system analysis functionality. The next step in the fingerprinting process will be determined by the product and vendor to make an educated guess as to the OS if it fails, then another test could be tried before failing out. From there it will attempt to identify defining characteristics of a specific device. However, as discussed earlier Android is the only operating system that is currently implemented.

Android Debug Bridge

The android debug bridge is included with the Android SDK [5]. It is used to allow easier debugging and analysis of connected Android devices. The ADB is a powerful tool for developers of applications for the Android OS. It also provides many useful things to a forensic examiner. The ADB works through three separate entities:

- Client: The simulator the user interacts
- Server: What manages the communication between the digital forensics and an android device.

- Daemon: what runs on the actual device and facilities the communication.

In order to create the fingerprint, I first looked at what commands that could be reasonably used to identify an individual phone. One of the first commands that was considered was the getprop command. This command provides many useful items about the device. After reviewing the properties several were selected to be used. The choice was based on research on the properties with respect to how difficult, if even possible to alter it. From this we determined that these properties where the most useful to creating a fingerprint:

- ro.build.display.id
- ro.build.version.incremental
- ro.product.cpu.abi
- ril.bt_macaddr or ro.bt.chipset
- ro.ril.MEID or ro.serialno

The first two were selected because many Android devices run different versions of official Android releases as well as custom ROMs installed on it. The important thing to consider in this case is that different versions have android have different capabilities, security mechanisms, and general layout. For example the recent change to the ext4 file system from yafs could create issues for low level data acquisition.

The other hardware options where used because changing those values will cause the phone to not work properly. They are also not easily changeable by the user without a high level of technical capability. Changing the model of cpu.abi would cause issue with the byte code the application java code is translated into, potentially making the device unstable. Finally, the Bluetooth MAC address and the MEID idea number cannot be changed by the end user. The only way that was ever found to be mentioned, would be to alter the MEID and would require the chips to be reprogrammed or the chips to be physically replaced. Neither of these things are easy to accomplish. Additionally, it could introduce issues with registering for service with a cellular provider.

The ro.bt.chipset and ro.serialno where added because as showed in the case study section there was one on device that did not return an MEID number and Bluetooth MAC address but it did have the serial number and Bluetooth chipsets so a change was made in the code to support the chance that that information may not be available. If those two alternatives are not included then the application simply ignores them and leaves the user to fill in the information with what they are able to determine from other methodologies.

The next step is to try to look at files that are unlikely to be altered by an individual user. In my case I used the proc/meminfo file to gather information about the type of memory that was available. This again is something that would be unlikely for the user to change and if it has changed most likely the someone has tampered with device, or as discussed earlier there is a chance that this is an indication that there is a hidden operating system, or cleverly hidden monitoring software on the device.

The advantage of performing these sorts of simple queries and low level of interaction to reduce the chance of accidently damaging the device. The other advantage is that if there is a fake operating system installed, the chance of these simple queries breaking the device is not very high.

Digital forensics Software Database

The digital forensics software compatibility database is designed to serve two purposes. At a larger level it will be distributed on a publicly available server to allow users around the world to have access to a single repository of information regarding various specifications and capabilities of different phones.

Additionally, it will also provide information about how well the devices interact with different forensics software packages. This will give investigators a quick reference guide to determine what they can accomplish to retrieve information off of the phone in a forensically sound manner. This will be established because the database will only be editable by people whom have licenses to use the software. This will help reduce the chance of inexperienced and malicious users from contaminating the database with false or incorrect information. The combination of this will make a difference in that the forensic examiner will now have an easy to use, concise, and hopefully fairly accurate technique to examine the device. Rather than relying on Google and potentially inaccurate internet forums.

The database is made up of a series of tables that create a system of information about a given phone model. The primary table is the “handset” table this contains the information for every device scanned into the system. It contains information such as the mac address, operating system information, processor, and any other information that can be found that may be useful for fingerprinting or gathering specific information for a device. These include tables for both the cellular radio and the wireless radios which were not utilized in our case study, but could be used for other operating systems. Finally, it uses a hash of the vendor id, product id, and operating system to link into the compatibility list table. This table contains the information about a particular model and how it interacts with various software suites and techniques. From this table, information can be displayed to the end user.

The database itself will reside on a server that can be accessed by anyone who has a license to the software. The users will also be able to edit data and add new data. The idea being, that by allowing a central repository for the knowledge and hopefully make it more easily searchable. This will also have the advantage because it should have a higher quality than doing a Google search.

The data that the users of software will input their techniques and that will be displayed to every other user that analyzes a similar device. These techniques will then have ability to be voted on by the users. This means that the software and different techniques that have the highest reliability and ease of use should be voted to the top of a given handset's results. This makes for a situation where there is a two-factor authentication for the validation of a given method. The first is that the software would only be distributed to individuals already working in the field. Reducing the odds of someone maliciously, or unknowingly posting bad information. Secondly the voting feature will mean that even techniques that are not correct or not reliable should not be among the first results.

By using the proposed software fingerprinting database the investigator will be able to quickly evaluate what options they have available as well if there are any major issues internally doing the digital forensics analysis that they investigators need to be aware of. By using this, they can determine if the analysis is something they can carry out or if they should hand the device off to a larger organization. The tradeoff being that by allowing a larger organization to handle it, they could greatly increase the time to get a result back, potentially creating an issue for the prosecutors with getting ready for a trial.

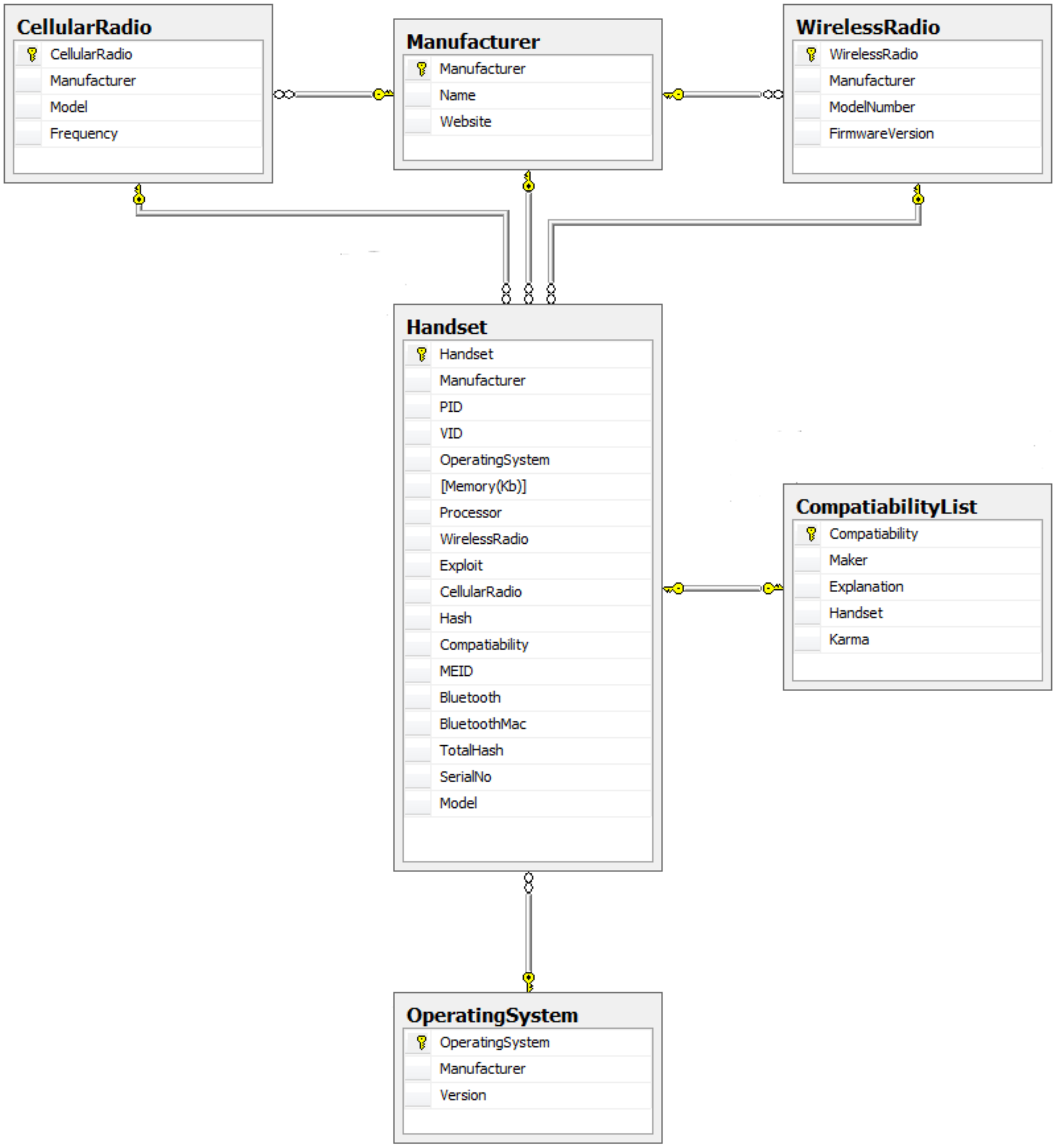

Figure 4-4

Diagram of the fingerprinting database

The issue of device compatibility with digital forensics tools

The large number of phones available on the market has created an issue with software packages being able to gather data from them. The issue arises from the concerns that many of the software packages do not provide full support for extracting data. The common definition of a package being compatible with a given phone is if it can access and obtain any one of the following:

- Contacts
- Recent calls
- Text Messages
- E-mails
- Applications
- Speed dial
- Browser History and Favorites

This obviously covers a lot of ground and allows for software manufacturers to claim compatibility with a high number of devices even if their usefulness for a particular phone is of dubious quality.

Additionally, given the sheer number mobile devices coming out on to the market testing each of them for compatibility and then making them compatible with the new devices is a monumental task for an agency to accomplish. This does not even address the different versions of software that could behave differently. For example, the recent upgrade to Encase to version 7 many of the previously used scripts to break. These changes may not have a specific effect on the actual digital forensics aspects but the change would have a definitive effect on a forensic examiner's ability to use it. The ability for a centralized publicly available database for forensic examiners to share their knowledge would prevent potential costly mistakes.

Chapter 5: Case Study

To test the viability of the software a sample of available Android phones was taken these included:

- Samsung Galaxy running Froyo
- Samsung Galaxy running Gingerbread
- HTC Incredible
- Asus Eee Transformer

During the testing of these phones we were able to determine that each phone was an individual device. We can see that each phone has a distinct in result which could be used to determine what type of techniques it could be then linked with in the software compatibility database to successfully determine what tools and techniques will allow it to work correctly. The largest challenge that was faced was that there is a chance that USB debugging is not always enabled; although, in this small subset, only one device did not have it enabled upon receipt. The lack of USB debugging would make the Android debug bridge generally unusable, but many syncing and streaming applications need it enabled so it is reasonable to think that many devices would have it enabled.

The phones where owned by various people who are known to the author so device specific information has been anonymized.

For example, when looking at a Samsung Galaxy Mesmerize phone running Gingerbread on US Cellular, we were able to recover the following information for saving and identification:

- VID: 04E8
- PID: 681C
- GINGERBREAD.EH09
- EH09
- 2013E01XXXXX
- 2013E01 XXXXX
- 355920 kB

The Samsung Galaxy running Froyo from US Cellular gave us the following:

- VID: 04E8
- PID: 681C
- FROYO.EE19
- EE19
- 2013E01XXXXX
- 2013E01XXXXX
- 355920 kB

Important note: The Galaxy phones where the same physical phone, however, running different versions of the OS show how the information is different even though phone is physically the same.

The third phone is a HTC Incredible from Verizon Wireless

- VID: 0BB4

- PID: 0C9E
- FRF91
- FRF91
- armeabi-v7a
- HT09XXXXXXXX
- Broadcom BCM4329-B1
- 423200 kB

The HTC phone was unique in that it did have either an MEID number or Bluetooth MAC address. This issue led to a change in how the system operated and added the ro.serialno and ro.bt.chipsetversion as options for fingerprinting. The fourth device was a Asus Eee Transformer gave us this information:

- VID: 0B05
- PID: 4E1F
- HTJ85B.US_epad-8.6.5.6-20110726
- US_epad-8.6.5.6-20110726
- armeabi-v7a
- 03700148XXXXXXXX

From this we can easily identify this phone by the type of device the version of operating system it is using (an important in Android where devices could potentially be running a number of OS versions) and also as an individual entity. From there the VID, PID, and operating systems are all combined to create a link to the digital forensics compatibility database. While the digital forensics database is currently empty due to lack of available equipment, filling it in can be easily accomplished.

In order to repeat these steps to get similar data it can be accomplished by using the Android debug bridge with the commands discussed earlier and reviewing the device manager.

Chapter 6: Future Work

This software only looks at the Android operating system. There are some issues in that it relies on USB Debugging enabled, however, many carriers are able to turn on USB Debugging remotely. However, while the fastest growing mobile operating system; there are still many other popular operating systems. So, the first step of future work is to expand the software suite out to other operating systems such as Apple’s iOS and RIM’s Blackberry OS. These are the next two most popular operating systems for mobile devices. The next step for furthering fingerprinting is to use it for a true security purpose is to add additional scanning of various aspects of the applications installed and changes in various aspects of the software to trigger alerts about the potential of being compromised.

Windows Phone 7

Windows Phone 7 is the newest Microsoft operating system for mobile devices. It was released initially in late 2010. It uses the Metro design language to develop apps. Unlike Android there is no comprehensive debug bridge readily available to developers. However there is a piece of software that is capable of gaining device information. This software is called the “Windows Phone Device Manager” and was released by Julien Schapman, a well-known Windows Phone developer, on the TouchXperience forum [12]. This piece of software purports to be able to gain detailed device information similar to what was accomplished using the Android Debug Bridge which implies that there is a way to gain such information. One potential drawback to Mr. Schapman's is that it requires the phone to be unlocked which is roughly equivalent to "rooting" on an Android device. However if that capability is required for the actual device information gather is necessary is unclear.

iOS

iOS in some ways would be the simplest operating system to determine information from. This is largely due to the fact that Apple produces relatively few variations on their devices, they release roughly three new versions of the iPhone every year. This means that by just using the product and vendor id's the software would acquire more information than one might with other devices. One issue that could arise is the lack of major developer tools on the Windows Operating System, potentially limiting the amount of effort that could be determined without spending a substantial amount of time and money developing it. That additional development would have a damaging effect on the idea of being able to provide a cheap and simple solution to law enforcement agencies. It is possible that the LibUSB .net library may be able to provide some useful information for creating a fingerprint, although may not be as detailed as what one can get using developer tools like the Android Debug Bridge.

Blackberry

Blackberries are arguably the most complicated of phones to analyze due to the fact that they were built with security in mind from the ground up making them very difficult to interface with in a way that lends itself to this kind of analysis or any forensics analysis. We can still gather several some basic information from the device on being plugged in. Like with iOS devices it is possible that using the LibUSB library gathering information from the Windows device information may prove useful for generating a fingerprint.

Chapter 7: Conclusion

This research introduces several novel ideas to streamline the processing of mobile devices and also accurately identifying them in a digital forensics environment. The software is designed to be a simple starting point which will provide forensic examiners a beginning point with which they can start the forensics examinations. The ability to know with a high degree of confidence, the effectiveness of a given piece of software or technique can save both the forensics examiners valuable time and also save their agencies money on spending of products they may not actually need. The end result of this is will be more cost and time effective production out of a digital forensics lab.

The ability to have a clue if something has been changed to result in a compromised device would prove invaluable to IT security and the intelligence community. This is where a detailed fingerprint would allow quick analysis to find out if someone has altered the device.

As discussed at the beginning of this article, there are many ways that this software could be used by a first responder and quickly analyze a mobile device and potentially save someone’s life.

References

1)" Run Android on Your Windows Mobile Phone - How-To Geek ." How-To Geek - Computer Help from your Friendly How-To Geek . N.p., n.d. Web. 30 Sept. 2011. <http://www.howtogeek.com/howto/20703/run-android-on-your-windows-mobile-phone/>.

2)" Android/Linux for CDMA Touch Pro 2 - PPCGeeks." PPCGeeks. N.p., n.d. Web. 30 Sept. 2011. <http://forum.ppcgeeks.com/android-tp2/113373-android-linux-cdma-touch-pro-2-a.html>.

3) "LibUsbDotNet 2.2.8 - Table of Content." LibUsbDotNet 2.0.0. N.p., n.d. Web. 30 Sept. 2011. <http://libusbdotnet.sourceforge.net/V2/Index.html>.

4)Sakr, Sharif. "Kirf -- Engadget." Engadget. N.p., n.d. Web. 30 Sept. 2011. <http://www.engadget.com/tag/kirf/>.

5)"Android Debug Bridge | Android Developers." Android Developers. N.p., n.d. Web. 30 Sept. 2011. <http://developer.android.com/guide/developing/tools/adb.html>.

6) "XRY Logical." Micro Systemation XRY. N.p., n.d. Web. 3 Oct. 2011. <http://www.msab.com/xry/xry-logical>.

7) "Cellebrite - Mobile Forensics and Data transfer solutions - UFED Physical Pro." Cellebrite - Mobile Forensics and Data transfer solutions. N.p., n.d. Web. 3 Oct. 2011. <http://www.cellebrite.com/forensic-products/forensic-products/ufed-physical-pro.html>.

8) "Cellebrite - Mobile Forensics and Data transfer solutions - UFED Phone Detective ." Cellebrite - Mobile Forensics and Data transfer solutions. N.p., n.d. Web. 3 Oct. 2011. <http://www.cellebrite.com/forensic-products/forensic-products/ufed-phone-detective.html>.

9) Jansen, Wayne, and Rick Ayers. *Guidelines on Cell Phone Forensics*. Gaithersburg, MD: National Institute of Standards and Technology, 2007. Print.

10) "Mobile Forensic Software Comparison Chart." *Paraben*. Paraben, n.d. Web. 14 Sept. 2011. <www.paraben.com/downloads/ds-comparison-chart.pdf>.

11) Schapman, Julien . "TouchXperience • View topic - Windows Phone Device Manager & TouchXperience betas." *TouchXperience • Index page*. N.p., n.d. Web. 9 Oct. 2011. <http://forum.touchxperience.com/viewtopic.php?f=30&t=783>.

12) Schapman, Julien . "TouchXperience • View topic - Windows Phone Device Manager & TouchXperience betas." *TouchXperience • Index page*. N.p., n.d. Web. 9 Oct. 2011. <http://forum.touchxperience.com/viewtopic.php?f=30&t=783>.

13) Brothers, Sam . "iPhone Tool Classification." *Welcome*. N.p., n.d. Web. 10 Oct. 2011. <http://www.appleexaminer.com/iPhoneiPad/ToolClassification/ToolClassification.html>.

14) Letaw, Lara, Joe Pletcher, and Kevin Butler. "1 Host Identification via USB Fingerprinting.” (2011): *http://ix.cs.uoregon.edu/~zephron/lpb11.pdf*. Web. 5 Sept. 2011.

15) "Logicube CellDEKÂ® - Cell Phone Data Extraction." *Logicube.com, hard drive duplication, copying hard drives & computer forensics*. N.p., n.d. Web. 15 Oct. 2011. <http://www.logicubeforensics.com/products/hd_duplication/celldek.asp>.

16) "Secure View Kit for Forensics-CP200-S20." *DataPilot Cell Phone Mobile Sync Software & Data Cables*. N.p., n.d. Web. 15 Oct. 2011. <http://www.datapilot.com/productdetail/253/producthl/Notempty>.

17) Song, Aly . "The latest fake iPhone | Reuters.com." *Business & Financial News, Breaking US & International News* | *Reuters.com*. Reuters, 10 Aug. 2011. Web. 18 Sept. 2011. <http://www.reuters.com/news/pictures/slideshow?articleId=USRTR2PTQG#a=2

18) "XRY Physical." *Micro Systemation XRY*. N.p., n.d. Web. 15 Oct. 2011. <http://www.msab.com/xry/xry-physical>.