FLiA

Forensic Investigation of iOS Apps

By

David Tromblay, Ersan Refaei, Robert Hinh

A Proposal Submitted to
The Faculty of the School of Information Technology
In Partial Fulfillment of the Requirements for
The Degree of Bachelor of Science
In Information Technology

University of Cincinnati
School of Information Technology
College of Education, Criminal Justice, and Human Services

April 2014

David Tromblay

Ersan Refaei

Robert Hinh

Professor Stockman

Date 4/18/14
Date 4/18/14
Date 4/18/14

Date
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section</td>
<td>i</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iii</td>
</tr>
<tr>
<td>Abstract</td>
<td>iv</td>
</tr>
<tr>
<td>1. Project Description and Intended Use</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Statement of the Problem</td>
<td>2</td>
</tr>
<tr>
<td>2. Proposed Course of Action</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Background</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Solution</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Deliverables</td>
<td>6</td>
</tr>
<tr>
<td>2.4 Target Audience</td>
<td>7</td>
</tr>
<tr>
<td>3. General Use Case Diagram</td>
<td>9</td>
</tr>
<tr>
<td>4. Tools</td>
<td>10</td>
</tr>
<tr>
<td>5. Budget</td>
<td>12</td>
</tr>
<tr>
<td>6. Timeline</td>
<td>14</td>
</tr>
<tr>
<td>6.1 Scrum Diagram</td>
<td>14</td>
</tr>
<tr>
<td>7. Proof of Concept</td>
<td>16</td>
</tr>
<tr>
<td>7.1 FliA Instruction Guide</td>
<td>16</td>
</tr>
<tr>
<td>7.2 FliA Website</td>
<td>19</td>
</tr>
<tr>
<td>7.3 FliA Documentation</td>
<td>22</td>
</tr>
<tr>
<td>8. Testing</td>
<td>23</td>
</tr>
<tr>
<td>8.1 Testing Introduction</td>
<td>23</td>
</tr>
<tr>
<td>8.2 Testing Conclusion</td>
<td>23</td>
</tr>
<tr>
<td>8.3 FliA Website Review</td>
<td>24</td>
</tr>
<tr>
<td>9. Conclusion and Recommendations</td>
<td>25</td>
</tr>
<tr>
<td>9.1 Conclusion</td>
<td>25</td>
</tr>
<tr>
<td>9.2 Recommendations</td>
<td>25</td>
</tr>
<tr>
<td>10. Works Cited</td>
<td>27</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1: General Use Case Diagram .......................................................................................... 9
Figure 2: Team Budget .................................................................................................................. 12
Figure 3: Budget to Reproduce Project ....................................................................................... 13
Figure 4: Project Timeline ........................................................................................................... 14
Figure 5: Jailbreaking iOS ........................................................................................................... 16
Figure 6: Installing & Configuring Cydia .................................................................................... 17
Figure 7: Tool Installation & Configuration ............................................................................... 18
Figure 8: FliA Website Home Page ............................................................................................ 19
Figure 9: FliA Website Contact Page ......................................................................................... 20
Figure 10: FliA Website Home Page Mobile .............................................................................. 207
Figure 11: FliA Website About Page Mobile .............................................................................. 21
Acknowledgements

We would like to thank our industry advisor Cardinal Solutions, especially Rusty Weneck, Eric Rolf, and Thomas Hoffman, for providing us with insight into investigating iOS apps, allowing us a constant medium for which to bounce ideas off of, and for providing us with funding for our project. Along with Cardinal Solutions, we would also like to thank Professor Stockman for acting as sounding board for ideas, as well as providing us with valuable input and guidance.
Abstract

The goal of the Forensic Investigation of iOS Apps project was to create a process for assessing the security of iOS devices, focusing extensively on the vulnerabilities within applications. The method used to accomplish this project was comprised of three separate parts. The first was to create an instruction guide that gives our readers the steps for performing a forensic investigation of iOS applications. The second being, a written report on the security vulnerabilities that our team found on iOS apps as a result of following the steps of our guide. And lastly, in order to bring our findings and knowledge to a larger audience, a Website was created to house all of the information from this project. This project provides our readers with the techniques, tools, and knowledge necessary in order to determine the security of iOS apps.
1. Project Description and Intended Use

1.1 Introduction

Over the past couple of decades, technology has been an integral part of our society, and as time progresses, humans will become even more dependent on technology as exponential advances continue to improve its abilities. This dependency on technology can be seen through its pervasiveness into almost every aspect of society; ranging from economics, to politics, to education, and even to individual’s personal lives. Its projected growth, and our dependency on it, can be put into perspective through Moore’s Law, which states that the “overall processing power for computers will double every two years” (Moore’s Law).

A contributing component of our growing dependency on technology is the growth of mobile computing, with mobile Internet traffic accounting for 15% of global Internet traffic, as of May 2013, and growing at a rate of approximately 1.5 times per year (Meeker and Wu). Further insight into the growth of, and transition towards, mobile computing can be gained through observing the global sales of companies that sell mobile devices. Specifically in the case of Apple, which is just one of the many companies that sell mobile devices; however, Apple’s mobile devices are the only ones that run iOS, one of the major mobile operating systems. The prevalence of iOS in the mobile device market is due in part to the number of iOS devices (e.g. iPhones and iPads) sold by Apple, which as of October 2013 has exceeded 700 million devices (Mathis). As well as in smartphone usage, where as of July 2013, iPhones accounted for a 40.7% market share of the total U.S. smartphone subscribers, which is a rise of 7.8% over 2012 (comScore Incorporated).
1.2 Statement of the Problem

This shift towards mobile computing makes the security of apps developed for mobile devices a major area of concern within information technology. Contributing to this concern is the total number of apps available to iOS devices from the App Store, which at the time of this writing, is estimated at over 760,000 apps (Monti). And although all iOS apps available on the App Store have been through Apple’s strict approval process, security testing of over 2,000 iOS apps “showed that nine out of 10 apps had serious vulnerabilities” (Messmer). An example of how insecure apps can make their way onto iOS devices, according to the research paper *Jekyll on iOS*, is through "Jekyll apps”. These “Jekyll apps”, “allow developers to reliably hide malicious behavior that would otherwise get their app rejected by the Apple review process”.

And once the app passes the review process, it becomes available on the store, then it is installed on an iOS device, where it can begin its malicious behavior (Wang, Chung and Lu).

Users of iOS devices know very little about the behavior of the apps that they use, as most apps simply do not disclose this information to the user. Therefore, behaviors such as, what data the app accesses, how it is stores information, and what network traffic it generates are all unknown to iOS users (Perlroth and Bilton). To further compound this lack of information, iOS apps can prompt users for permission to access contact, location, and other personal information, but again most do not disclose why they require access. In addition, these permissions that are granted to apps can be abused, as was found in the research publication *Malicious Android Applications in the Enterprise*, where “permissions to access personal, location, and contact information are abused by most apps that requested such permissions” (Wei, Gomez and Neamtiu 252). This same permission abuse scenario is possible in the iOS platform, with users being unaware as to when an app will access this permitted information, and for what purpose.
To summarize, the only app behavior information iOS users know, is that which the app chooses to disclose along with what information is contained on the App Store.

In stark contrast to iOS apps, Android apps provide their users with more information about what data and services the app will access on the device. Due to this, Android users have a better idea of the behavior of the apps that they use. This lack of transparency into iOS apps, from the user’s perspective, makes it possible for iOS users to use an app that compromises the security of their iOS device without their knowledge. Some examples of possible vulnerabilities apps can contain are: storing data in an unprotected way, transmitting and receiving data from some unknown server, and lack of data validation (input/output) (Security Learn).

This problem of application security is concern enough for everyday users of iOS devices, as the information on their device can be jeopardized by the various vulnerabilities an app may contain. However, with Bring Your Own Device (BYOD), where employees are able to use computation and communication devices they choose in order to increase their productivity and mobility at work; having an iOS device with an insecure app can open up a business’s network infrastructure to unseen vulnerabilities (Anderson 5). The impact of which is significantly greater than if an iOS device is used strictly for personal use, as a business’s information is generally more valuable, as it contributes to the overall success of the business as a whole.

The overall scope of this problem of iOS app security, from personal use all the way to business use, and the effect that it can have on the information contained and accessible to iOS devices made it a vital problem to address and solve.
2. Proposed Course of Action

2.1 Background

The primary area of emphasis for the FIIA project was systems/networking, with a secondary emphasis on databases, because in regards to the majority of mobile applications:

- Systems give the representation of how different components work in conjunction with each other (i.e. how apps share data).
- Networks give mobile applications the ability to transmit information to servers, other devices, and other applications.
- Databases give mobile applications a location to store their information.

From the definitions above, knowledge and understanding of systems, networks, and databases played a pivotal role in this project.

There are currently a number of different options out there for iOS users to determine the security of the apps they use. They can search the Internet for security reviews, look through the App Store for user reviews, or try to find a website that describes how to analyze an iOS app, just to name a few. However, there are flaws in each of these options for determining the security of an iOS app. First, since most apps are updated regularly, any information found on the Internet or from a user review on the App Store can be referencing a previous version of an app, which means the information could be of little relevance to the current version of the app. Second, user reviews on the App Store lack credibility, as anyone with an Apple account can review an app. And third, websites that describe how to analyze iOS apps may not cover a particular area of the investigation process, in which case another website is needed in order to fill in that gap. This can lead to a disjointed investigation process in terms of the methods and
tools that are used, which can make the forensic investigation process more difficult for iOS users to follow.

2.2 Solution

The suggested solution to raise awareness about vulnerable iOS apps is to establish a step-by-step instruction guide and vulnerability document that helps users perform their own forensic analysis on iOS apps. In addition to the instruction guide and vulnerability documentation, a centralized website where iOS users can download our instruction guide will further raise awareness of vulnerable iOS apps.

The FIiA project solves the problem of iOS app security as outlined below for the four core components of our project:

1.) A centralized instruction set that shows readers how to perform a forensic investigation on the iOS apps that they use. This includes a detailed list of tools that can be used, how to configure them, and how to use them for other forensic purposes.

2.) Identify and specify common vulnerabilities to look for in iOS applications, with explanations as to why they deserve attention. In other words, what types of data the app will have access to, how the app uses the data, how much the app affects the hardware's performance, etc.

3.) Identify and describe the iOS app best practices that make up a “secure” application (i.e. what you would ideally see when performing a forensic investigation of an iOS app).

4.) Documentation of vulnerabilities our team finds on own forensic investigation of popular free and paid iOS apps using the instruction guide we created. This documentation provides readers with examples of actual forensic investigation findings.
2.3 Deliverables

The deliverables of the FIiA project are threefold: the FIiA Instruction Guide, the FIiA Documentation, and the FIiA Website. All of which were derived from the four core components of our solution stated above. All three deliverables were designed with the purpose of being used in conjunction with one another to create a unified process for the Forensic Investigation of iOS Apps.

The first deliverable is the FIiA Instruction Guide which provides users with step-by-step instructions, supplemented by appropriate screenshots, on how to go about performing a forensic investigation of iOS apps. It includes a detailed list of tools that are used throughout the FIiA Instruction Guide, the purpose of each tool, links to each tool, and instructions on how to install and configure them.

The second deliverable is the FIiA Documentation which houses our team’s forensic investigation findings of popular free and paid iOS apps. It details all the information regarding what iOS app was investigated, such as: the purpose of the app (e.g. a popular social networking app) the version of the app, and the date the app was investigated. The name of the app is intentionally left out due to legal reasons, as recommended to our team by Dr. Wei, but will be disclosed to Professor Stockman. It then describes and categorizes any and all vulnerabilities found in the app into the following four categories: Network vulnerabilities, Data vulnerabilities, Access vulnerabilities, and Code vulnerabilities. And finally the FIiA Documentation summarizes the vulnerabilities we found in relation to the security of the app being investigated based off of our findings.

The third deliverable is the FIiA Website which houses all of the data from the FIiA project. Specifically, it contains the FIiA Instruction Guide, the FIiA Documentation, and
explains the purpose of the FliA project. It also allows users to send emails to our team with suggestions of iOS apps for our team to perform forensic investigations on, as well as any other recommendations related to the project. The overall rationale for this deliverable being, that by creating a website we can disseminate our findings and knowledge from the FliA project to a much greater audience, rather than to just the University of Cincinnati community through our FliA project report. Additionally, the FliA Website helps to spread the awareness of the problem of iOS app security.

As a whole, the three deliverables of the FliA project enable users to: learn about the forensic investigation of iOS apps, perform their own forensic investigations of iOS apps, and suggest iOS apps that our team should perform forensic investigations on. It is our goal to use these three deliverables to deliver an easy to use process of investigating iOS applications.

2.4 Target Audience

The target audience for this solution falls into two distinct categories. In the first group are the technical users who have some background in mobile security and, most importantly, feel confident in performing a forensic investigation on their own. And in the second group are the non-technical users who are not confident enough in their abilities to perform their own forensic analysis, but are interested in learning more about the results of a forensic analysis and iOS app security in general.

For both categories of the target audience, our solution assumes that the users know how to read and know how to navigate a website using a computer; skills which would be sufficient for them to learn about iOS app security, forensic investigations, and the results of the forensic investigations. However, for the technical users who would like to perform their own forensic investigation, we assumed that they know how to operate their iOS device (which includes a
plethora of functions, such as downloading apps, starting apps, etc.) and they know how to download and install applications onto a computer. It would be ideal for technical users to have at least a basic knowledge of system security, so that they can better understand the purpose of each of the investigation processes as well as the vulnerabilities to look for during an investigation. However, it is not necessarily required as the FliA Instruction Set and the FliA Documentation describe to users the purpose of each investigation process and the vulnerabilities users should look for.
3. General Use Case Diagram

Below is a diagram (shown as Figure 1) of the general use case that most of the processes followed. It starts with the user and an iOS device, which connects to the service provider, either wirelessly or through service towers. Due to some iOS devices being mobile phones or tablets, the service provider can be either an ISP (Internet Service Provider) or a cellular provider. Either way, it uses the service provider’s infrastructure to connect to the internet. The iOS device may be used from within a company’s network infrastructure, in which case the business’s servers could be accessed before the device reaches the Internet. From the Internet, the iOS device will then connect to the application developer’s infrastructure, which contains their servers, services, databases, etc.

![General Use Case Diagram](image-url)
4. Tools

Below are a list of tools that we used in our FliA Instruction Guide and Documentation. We chose most of these tools based on the fact that they are either freeware or come with a free trial period. This significantly reduced the cost of our project, as we will detail later in section 5.

**plist Editor Pro** – plist Editor Pro is a software application that shows what data is on a phone, as well as app information such as the plists that might be on the phone (VOWSoft). Property Lists (plists) are where some app developers write code to save settings and configurations that the user may have set on their iOS device. In addition they can also store usernames and passwords.

**iExplorer** – iExplorer is a software application that allows users to look into the phone to see the apps that are installed on the phone, and allows for users to perform a quick analysis without having to jailbreak their iOS device, which removes encryption off the iOS device. With this tool, users can look at their apps before performing a jailbreak so that they can get a deeper view into the app (Macroplant).

**WinSCP** – WinSCP is an application that gives users the ability to SSH (secure shell) into an electronic device (WinSCP).

**PuTTY** – PuTTY is an application that emulates a terminal window and is comparable to WinSCP (PuTTY).

**Wireshark** – Wireshark is a free software application that allows users to trace packets across a network, and is the world’s foremost network protocol analyzer. We believe that it is the best tool for network analysis as it is the standard tool used across many industries, and it allows for the capture of any packets that are running through a network. With it we can analyze packets and see what kind of data is being sent to and from the iPhone (Wireshark).
**Burp Suite** – Burp Suite is a free software application that allows users to capture packets, analyze packet content, resend packets, and deny packets (PortSwigger).

**Ettercap** – Ettercap is one of the world’s best programs to perform man-in-the-middle-attacks (MiTM). For development purposes, application developers tend to test their applications by disabling the SSL chain security function, but sometimes forget about enabling the function before it heads back into production. When this happens the application that has the SSL vulnerability could be used later in a MiTM attack. Kali Linux is a security toolkit has Ettercap installed as part of the operating system (Kali Linux).

**Kali Linux** – Kali Linux is a Linux based security toolkit that specializes in penetration testing, and includes Ettercap (Kali Linux).

**Clutch** – Clutch is an application that needs to be installed on the phone after the iOS device has been jailbroken. This application is used to help decompile the code in iOS apps. This is important because iOS apps are loaded into a virtual sandbox where Clutch can gather all the packets that are being used by the application and perform an analysis on the unencrypted app. Basically, Clutch runs the app and creates snapshots of what the app is doing so that users can see the app in plain text and be able to look at its code.

**Hopper** – Hopper is a reverse engineering application that allows users to disassemble, decompile, and debug iOS executables (Cryptic Apps SARL).
5. Budget

As a team, for us to efficiently perform a forensic investigation, we needed two iOS devices and two computers, both of which we supplied ourselves. As for the iOS apps, we did not want to limit ourselves to only using free applications for two reasons. The first is that to believe that our target users only use free applications is obstinate. The second is that we were interested to see the differences between paid applications and their free counterparts, and their effects on the iOS platform.

The technologies and hardware that were required for the FliA project are listed in the budget table, shown below as Figure 2. Between using our own tools, which is denoted by (supplied by team), and with the generous support from our industry advisor, the Cardinal Solutions Group, which is denoted by (supplied by funds), the cost to us for the FliA project is $0.00.

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Cost</th>
<th>Cost to us</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS Device (iPhone)</td>
<td>2</td>
<td>$200 +</td>
<td>$0.00 (supplied by team)</td>
</tr>
<tr>
<td>Computer</td>
<td>2</td>
<td>$300 +</td>
<td>0.00 (supplied by team)</td>
</tr>
<tr>
<td>iOS Apps</td>
<td>~10</td>
<td>$0 - $50 +</td>
<td>0.00 (supplied by funds)</td>
</tr>
<tr>
<td>iExplorer</td>
<td>2</td>
<td>$49.99</td>
<td>0.00 (supplied by funds)</td>
</tr>
<tr>
<td>Hopper</td>
<td>2</td>
<td>$45.00 - $59.00</td>
<td>0.00 (supplied by funds)</td>
</tr>
<tr>
<td>plist Editor Pro</td>
<td>2</td>
<td>$29.95</td>
<td>0.00 (supplied by funds)</td>
</tr>
<tr>
<td>Burp Suite</td>
<td>2</td>
<td>Freeware</td>
<td>0.00</td>
</tr>
<tr>
<td>Wireshark</td>
<td>2</td>
<td>Freeware</td>
<td>0.00</td>
</tr>
<tr>
<td>Clutch</td>
<td>2</td>
<td>Freeware</td>
<td>0.00</td>
</tr>
<tr>
<td>Ettercap</td>
<td>2</td>
<td>Freeware</td>
<td>0.00</td>
</tr>
<tr>
<td>Kali Linux</td>
<td>2</td>
<td>Freeware</td>
<td>0.00</td>
</tr>
<tr>
<td>WinSCP</td>
<td>2</td>
<td>Freeware</td>
<td>0.00</td>
</tr>
<tr>
<td>Putty</td>
<td>2</td>
<td>Freeware</td>
<td>0.00</td>
</tr>
<tr>
<td><strong>Total Cost</strong></td>
<td></td>
<td></td>
<td><strong>$0.00</strong></td>
</tr>
</tbody>
</table>

*Figure 2: Team Budget*

Note: The + symbol means the cost can exceed the value listed. And the ~ symbol stands for approximately (e.g. ~10, which is read as, approximately 10)
There can be a variance in the cost of each item due to: prices can change over time, some items can be purchased new or used, and there can be various item options within a category (e.g. iOS device and computer). Therefore, it is extremely difficult to provide an exact cost for each item, and as such, the cost values for the iOS Device and Computer categories are approximate baseline values for those items. And the cost value for the iOS Apps category reflects the fact that apps can either be, free or paid, thus a range of values is given.

The budget table, shown below as Figure 3, is a compiled budget that details the cost required to duplicate our project in order to show our audience what each item could cost them if they wanted to implement our project on their own. The total cost estimate was calculated by adding up all the values listed in the Cost column, with the highest range cost value being used for rows that used a range of cost values (i.e. iOS apps and Hopper).

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS Device (iPhone)</td>
<td>1</td>
<td>$200 + (per device)</td>
</tr>
<tr>
<td>Computer</td>
<td>1</td>
<td>300 + (per system)</td>
</tr>
<tr>
<td>iOS Apps</td>
<td>~10</td>
<td>0 - 50 +</td>
</tr>
<tr>
<td>iExplorer</td>
<td>1</td>
<td>0 (Free Trial)</td>
</tr>
<tr>
<td>Hopper</td>
<td>1</td>
<td>0 (Free Trial)</td>
</tr>
<tr>
<td>plist Editor Pro</td>
<td>1</td>
<td>0 (Free Trial)</td>
</tr>
<tr>
<td>Burp Suite</td>
<td>1</td>
<td>0 (Freeware)</td>
</tr>
<tr>
<td>Wireshark</td>
<td>1</td>
<td>0 (Freeware)</td>
</tr>
<tr>
<td>Clutch</td>
<td>1</td>
<td>0 (Freeware)</td>
</tr>
<tr>
<td>Ettercap</td>
<td>1</td>
<td>0 (Freeware)</td>
</tr>
<tr>
<td>Kali Linux</td>
<td>1</td>
<td>0 (Freeware)</td>
</tr>
<tr>
<td>WinSCP</td>
<td>1</td>
<td>0 (Freeware)</td>
</tr>
<tr>
<td>Putty</td>
<td>1</td>
<td>0 (Freeware)</td>
</tr>
<tr>
<td><strong>Total Cost Estimate</strong></td>
<td></td>
<td>~$550.00 +</td>
</tr>
</tbody>
</table>

Note: The + symbol means the cost can exceed the value listed. And the ~ symbol stands for approximately (e.g. ~10, which is read as, approximately 10)
6. Timeline

Below in Figure 4 is our timeline for the project over the course of the senior design.

<table>
<thead>
<tr>
<th>August</th>
<th>September</th>
<th>October</th>
<th>November</th>
<th>December</th>
<th>January</th>
<th>February</th>
<th>March</th>
<th>April</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project decision</td>
<td>Project research</td>
<td>Create/Improve foundation report</td>
<td>Create/Improve analytics models</td>
<td>Analyze list of applications</td>
<td>Create centralized website</td>
<td>Finalize and submit report</td>
<td>Create mobile application</td>
<td>Presentation and tech expo</td>
</tr>
</tbody>
</table>

Figure 4: Project Timeline

6.1 Scrum Diagram

In addition, Figure 5 below shows a generalized overview of each major component of the project. The six components are (not in any specific order): Project decision, documentation, instruction set, creating web/mobile application, analyzing iOS applications, and presentation. The diagram also has these components labeled within the semi-circle at the top of each process. The semi-circle at the bottom of each process is the ideal end result that we aimed to achieve by completing the process. The diagram below is related to the scrum methodology, which emphasizes iterations within the product development process (in our case project development) in order to adapt to changes that may arise. The idea being, that by adapting to change throughout each part of the development process the final product/project is more successful than if a waterfall methodology were used (Scrum.org).
Figure 5: Scrum Diagram
7. Proof of Concept

7.1 FIiA Instruction Guide

The figures below are sections of our FIiA Instruction Guide that serve as a proof of concept.

1. Jailbreaking iOS

1.1 iPhone Jailbreaking Intro

Jailbreaking is the process of bypassing the restrictions on iOS devices in order to install apps and tweaks not approved by Apple. By jailbreaking an iOS device we are breaking and unencrypting the device’s partition, which allows us to have access to the hard drive in the iPhone. By default, iPhones come with two partitions out of the box: one partition which has the OS and is encrypted, and the other partition which is used for the user data. Once the iPhone has been jailbroken, it will allow us access to look at the databases that applications are writing to.

1.2 Tools & Files to Download

Download these tools and files to an accessible location on your computer. Only the Jailbreak for your iPhone’s iOS version needs to be downloaded (i.e. if your iPhone is running iOS 6.x, only download Jailbreak iOS 6.x).

- **.ipsw file** – iPod/iPhone/iPad Software file that contains a full version of iOS. This type of file can be downloaded from [getiOS](http://www.getios.com/) based on the iOS device you have.

  [getiOS](http://www.getios.com/)

- **Jailbreak iOS 7.x** – Tool used to jailbreak devices running iOS 7.x. Can be run on Mac or Windows.

  [Evasi0n](http://evasi0n.com/)

- **Jailbreak iOS 6.x** – Tool used to jailbreak devices running iOS 6.x. Can be run on Mac, Windows, or Linux. When you navigate to the link below, select values from the Operating System, Software (Choose [Redsn0w](http://www.redsn0w.us/2010/03/download-direct-links-jailbreak-guides.html)), and Version dropdowns before clicking the Download Selected Tool button.

  [Redsn0w](http://www.redsn0w.us/2010/03/download-direct-links-jailbreak-guides.html)

*Figure 5: Jailbreaking iOS*
1.5 Installing & Configuring Cydia

Cydia is an app that gets installed when jailbreaking an iOS device. It is alternative App Store which offers apps and tools not available in the Apple App Store. We will use Cydia to download other apps and tools onto the iOS device, which are necessary for certain forensic investigation processes.

1. On the iPhone we need to locate the Cydia app (shown in Figure 14).

![Figure 14: Cydia App](image)

2. Ensure the iPhone is connected to Wi-Fi so that Cydia can update and pull down all the new packages that it needs.

![Figure 6: Installing & Configuring Cydia](image)
2. Tool Installation & Configuration

The tools in this section are applications which are needed for most of the subsequent forensic investigation sections. They can be thought of as the core tools needed to perform a forensic investigation of iOS apps.

2.1 Tools to Download

Download these tools to an accessible location on your computer.

iExplorer – A software application that allows users to look into the phone to see the apps installed on the phone, and to perform a quick analysis without having to jailbreak their iOS device, which removes encryption from the iOS device. With this tool, users can look at their apps before performing a jailbreak so that they can get a deeper view into the app (Macroplant).

http://www.macroplant.com/iexplorer/

plist Editor Pro – A free software application that shows what data is on a phone, as well as app information such as the plists that might be on the phone (VOWSoft). Property Lists (plists) are where some app developers write code to save settings and configurations that the user may have set on their iOS device. In addition, plists can also store usernames and passwords.

http://www.icopybot.com/download.htm
7.2 FliA Website

The figures below are screenshots of our FliA Website (www.fiia.info) that serve as a proof of concept.

Figure 8: FliA Website Home Page
Figure 9: FliA Website Contact Page
Mission Statement

Our goal with the Forensic Investigation of iOS Applications project was to create a unified framework for iOS Forensics. Our approach to tackling this area was...

About the Forensic Investigation of iOS Applications Project

Background

This project was done in connection with the University of Cincinnati and their Information Technology program. This senior design project was developed over two semesters starting from the project's inception, through development, and finally the production of three core deliverables: the FliA Instruction Guide, the...

Figure 10: FliA Website Home Page Mobile
Figure 11: FliA Website About Page Mobile
The figures below are screenshots of our FIiA Documentation that serve as a proof of concept.

App Name ######  Version #2.1.1  Price free

<table>
<thead>
<tr>
<th>Types</th>
<th>Vulnerabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Messages send in plain text</td>
</tr>
<tr>
<td>Code &amp; Access</td>
<td>Username/Password, location logged in from (before asking for permission)</td>
</tr>
<tr>
<td>Data</td>
<td>Unencrypted information in keychain data</td>
</tr>
</tbody>
</table>
8. Testing

8.1 Testing Introduction

The goal of our testing was to ensure high usability within all sections of our project. Since this project was intended to instruct users on how to investigate iOS apps, the user-friendliness of each deliverable contributed to the success of the project as a whole. Our testing had three main sections to review: the FliA Instruction Guide, the FliA Documentation, and the FliA Website. Each section of testing had its own unique areas of interest that needed to be tested. For the FliA Instruction Guide, it needed to be step-by-step with screenshots to ensure high usability, as well as the inclusion of relevant links to the tools that were used to further simplify the setup of the investigation environment. For the FliA Documentation, it had to represent our findings in a detailed breakdown in regards to each of the different categories of vulnerabilities. For the FliA Website, it had to be hassle-free when: navigating the site, sending emails to our team, and accessing our FliA Instruction Guide. The testing plan that we implemented had every member of our team review each item and comment on any changes that needed to be made. Furthermore, we enlisted the help of our industry advisors to help us review each item.

8.2 Testing Conclusion

The primary testers for our plan were each team member, along with our industry advisors. By using our FliA Instruction Guide to develop the FliA Documentation we ensured the guide’s usability and thoroughness, while confirming that each vulnerability section of our FliA Documentation was being addressed.
Other users that helped with our testing were primarily those who were willing to help by reviewing each of the three items, to verify that they were straightforward and have high usability.

8.3 FliA Website Review

As a team we were not concerned with the testing of the FliA website in regards to its functionality, as we had never experienced an issue that would cause concern. Throughout its development we had been testing each button and form function on the site to ensure that it was working as expected. There were only two areas of interest that we focused our testing on once the website went live, the sending of emails and the navigation to our project’s Facebook page. Both of these items were simple and quick to test, requiring little testing time and effort.
9. Conclusion and Recommendations

9.1 Conclusion

From the beginning of this project, iOS users could be subjected to security vulnerabilities in the form of insecure iOS Applications. With this lack of iOS app behavior disclosure, we sought to develop a process for which iOS users could use to perform their own investigation to determine the security of the iOS apps they use. This was the original reason for the Forensic Investigation of iOS Apps project and lead to the development of our three core deliverables: FliA Instruction Guide, FliA Documentation, and the FliA Website. With the development and implementation of these deliverables, we feel that our solution for this problem of iOS app security will be able to provide the processes and knowledge necessary for iOS users to perform their own iOS app investigations.

9.2 Recommendations

The primary goal of the FliA project was to bring iOS users together and provide them with the resources necessary to perform security investigations. In order to keep spreading knowledge about iOS app security, the FliA team will need to continue engaging current and future iOS users. While we have already done a good job of this so far, we must be sure to continue spreading the knowledge we have found in our project after senior design is over. We will continue this engagement through our FliA Website and by our social media mediums.

The FliA team has realized something worth noting throughout this senior design project. We have discovered through our investigation of iOS apps how easy it is for users to surrender their data to applications without even knowing it. A fair number of apps we investigated do not do a great job of securing and maintaining user data. If iOS users were educated on what apps
can do, even just a little bit, we feel there would be a significant decrease in the number of insecure apps on iOS devices.
10. Works Cited


