Chapter No. 2
"Understanding the Internals of iOS Devices"
In this package, you will find:

A Biography of the authors of the book

A preview chapter from the book, Chapter NO.2 "Understanding the Internals of iOS Devices"

A synopsis of the book’s content

Information on where to buy this book

About the Authors

**Satish Bommisetty** is a security analyst working for a Fortune 500 company. His primary areas of interest include iOS forensics, iOS application security, and web application security. He has presented at international conferences, such as ClubHACK and C0C0n. He is also one of the core members of the Hyderabad OWASP chapter. He has identified and disclosed vulnerabilities within the websites of Google, Facebook, Yandex, PayPal, Yahoo!, AT&T, and more, and is listed in their hall of fame.

I would like to thank everyone who encouraged me while producing this book, especially my wife for her great support.

For More Information:

Rohit Tamma is a security analyst working for a Fortune 500 company. His interests lie in mobile forensics, Android application security, and web application security. He is experienced in performing vulnerability assessments and penetration testing of a range of applications, including web and mobile applications. He lives in Hyderabad, India, where he spends time with his parents and friends.

I would like to thank everyone who encouraged me while I was authoring this book, especially my parents and my friends who offered their support in every way they could. Special thanks to Satish Bommisetty, my colleague, co-author of this book, who mentored me all the way through with his valuable suggestions.

Heather Mahalik is the Mobile Exploitation Team Lead at Basis Technology and the Course Lead for the SANS Smartphone Forensics course. With over 11 years' experience in digital forensics, she currently focuses her energy on mobile device investigations, forensic course development and instruction, and research on smartphone forensics.

Prior to joining Basis Technology, Heather worked at Stroz Friedberg and as a contractor for the U.S. Department of State Computer Investigations and Forensics Lab. She earned her Bachelor's degree from West Virginia University. She has authored white papers and forensic course material, and has taught hundreds of courses worldwide for law enforcement, Government, IT, eDiscovery, and other forensic professionals focusing on mobile devices and digital forensics.

There are a lot of people to whom I owe my deepest gratitude. This book is for my husband, who always encourages me to try harder and strive to be one step ahead. This book is also for Jack, who would sleep so that mama could write, and my dad and mother-in-law for always supporting me. Professionally, this book is for Brian Carrier, Eoghan Casey, Terrance Maguire, Rob Lee, and Shawn Howell for getting me addicted to this trade and providing me with the opportunities to better myself. I would also like to thank my co-workers, who have taught me patience, kept a smile on my face, and helped me learn more about forensics than most would deem required. You guys are the best!

For More Information:
Practical Mobile Forensics

The exponential growth of mobile devices has revolutionized many aspects of our lives. In what is called the Post-PC era, smartphones are engulfing desktop computers with their enhanced functionality and improved storage capacity. This rapid transformation has led to increased usage of mobile handsets across all sectors.

Despite their small size, smartphones are capable of performing many tasks—sending private messages and confidential e-mails, taking photos and videos, making online purchases, viewing salary slips, completing banking transactions, accessing social networking sites, managing business tasks, and more. Hence, a mobile device is now a huge repository of sensitive data, which could provide a wealth of information about its owner. This has in turn led to the evolution of mobile device forensics, a branch of digital forensics that deals with retrieving data from a mobile device. Today, there is a huge demand for specialized forensic experts, especially given the fact that the data retrieved from a mobile device is admissible in court.

Mobile forensics is all about utilizing scientific methodologies to recover data stored within a mobile phone for legal purposes. Unlike traditional computer forensics, mobile forensics has limitations when obtaining evidence due to rapid changes in the technology and the fast-paced evolution of mobile software. With different operating systems and a wide range of models being released into the market, mobile forensics has expanded over the last 3-4 years. Specialized forensic techniques and skills are required in order to extract data under different conditions.

This book takes you through the challenges involved in mobile forensics and practically explains detailed methods on how to collect evidence from different mobile devices with the iOS, Android, BlackBerry, and Windows mobile operating systems.

The book is organized in a manner that allows you to focus independently on chapters that are specific to your required platform.

For More Information:
What This Book Covers

Chapter 1, Introduction to Mobile Forensics, introduces you to the concept of mobile forensics, core values, and its limitations. The chapter also provides an overview of practical approaches and best practices involved in performing mobile forensics.

Chapter 2, Understanding the Internals of iOS Devices, provides an overview of the popular Apple iOS devices, including an outline of different models and their hardware. The book explains iOS security features and device security and its impact on the iOS forensics approach. The chapter also gives an overview of the iOS file system and outlines the sensitive files that are useful for forensic examinations.

Chapter 3, Data Acquisition from iOS Devices, covers various types of forensic acquisition methods that can be performed on iOS devices and guides you through preparing your desktop machine for forensic work. The chapter also discusses passcode bypass techniques, the physical extraction of devices, and different ways that the device can be imaged.

Chapter 4, Data Acquisition from iOS Backups, provides a detailed explanation of different types of iOS backups and details what types of files are stored during the backup. The chapter also covers logical acquisition techniques to recover data from backups.

Chapter 5, iOS Data Analysis and Recovery, discusses the type of data that is stored on iOS devices and the general location of this data storage. Common file types used in iOS devices, such as plist and SQLite, are discussed in detail so you understand how data is stored on the device, which will help forensic examiners to efficiently recover data from these files.

Chapter 6, iOS Forensic Tools, provides an overview of the existing open source and commercial iOS forensics tools. These tools differ in the range of mobile phones they support and the amount of data that they can recover. The chapter describes the advantages and limitations of these tools.

Chapter 7, Understanding Android, introduces you to the Android model, file system, and its security features. It provides an explanation of how data is stored in any android device, which will be useful while carrying out forensic investigations.

Chapter 8, Android Forensic Setup and Pre Data Extraction Techniques, guides you through the Android forensic setup and other techniques to follow before extracting any information. Screen lock bypass techniques and gaining root access are also discussed in this chapter.

For More Information:
Chapter 9, *Android Data Extraction Techniques*, provides an explanation of physical, file system, and logical acquisition techniques to extract information from an Android device.

Chapter 10, *Android Data Recovery Techniques*, explains the possibilities and limitations for data recovery on Android devices. This chapter also covers the process to reverse engineer Android applications to unearth crucial information.

Chapter 11, *Android App Analysis and Overview of Forensic Tools*, covers various available open source and commercial tools, which are helpful during forensic examination of Android devices.

Chapter 12, *Windows Phone Forensics*, provides a basic overview of forensic approaches when dealing with Windows Phone devices.

Chapter 13, *BlackBerry Forensics*, provides forensic approaches to include acquisition and analysis techniques when dealing with BlackBerry devices. BlackBerry encryption and data protection is also addressed.

For More Information:

As of September 2013, Apple had sold more than 550 million iOS devices (170 million iPads and 387 million iPhones) according to released sales records. While iOS is the leading operating system for tablets worldwide, Android continues to be the leading operating system for smartphones worldwide. The following screenshot represents the worldwide mobile/tablet operating system share from 2013 to 2014 according to https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=9&qpcustomb=1:

For More Information:
Regardless of the statistics, if you are a forensic examiner, chances are you will need to conduct an examination of an iOS mobile device.

In order to perform a forensic examination on an iOS device, the examiner must understand the internal components and inner workings of that device. Developing an understanding of the underlying components of a mobile device will help the forensic examiner understand the criticalities involved in the forensic process, including what data can be acquired, where the data is stored, and what methods can be used to access the data from that device. So, before we delve into the examination of iOS devices, it is necessary to know the different models that exist and their internals.

This book primarily focuses on the iPhone and forensic techniques associated with it. However, the same techniques may be applied to other Apple devices, such as the iPod Touch, iPad, and Apple TV.

### iPhone models

The iPhone is among the most popular mobile phones on the market. Apple released the first generation iPhone in June 2007. Ever since the first release, the iPhone has gained a lot of popularity due to its advanced functionality and usability. The introduction of the iPhone has redefined the entire world of mobile computing. Consumers started looking for faster and more efficient phones. Various iPhone models exist now with different features and storage capabilities to serve the consumer requirements. The following table lists all the iPhone models and its initial iOS versions. With the iPhone, individuals can access e-mail, take photos and videos, listen to music, browse the Internet, and do much more. Furthermore, endless applications are available for download to extend the standard capabilities that exist on the iPhone.

<table>
<thead>
<tr>
<th>Device</th>
<th>Model</th>
<th>Initial OS</th>
<th>Internal name</th>
<th>Identifier</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone 2G</td>
<td>A1203</td>
<td>iPhone OS 1.0</td>
<td>M68AP</td>
<td>iPhone 1,1</td>
<td>June 2007</td>
</tr>
<tr>
<td>iPhone 3G</td>
<td>A1241</td>
<td>iPhone OS 2.0</td>
<td>N82AP</td>
<td>iPhone 1,2</td>
<td>July 2008</td>
</tr>
<tr>
<td>iPhone 3G (china)</td>
<td>A1324</td>
<td>iPhone OS 2.0</td>
<td>N82AP</td>
<td>iPhone 1,2</td>
<td>July 2008</td>
</tr>
<tr>
<td>iPhone 3GS</td>
<td>A1303</td>
<td>iPhone OS 3.0</td>
<td>N88AP</td>
<td>iPhone 2,1</td>
<td>June 2009</td>
</tr>
<tr>
<td>iPhone 3GS (china)</td>
<td>A1325</td>
<td>iPhone OS 3.0</td>
<td>N88AP</td>
<td>iPhone 2,1</td>
<td>June 2009</td>
</tr>
<tr>
<td>iPhone 4 - GSM</td>
<td>A1332</td>
<td>iOS 4.0</td>
<td>N90AP</td>
<td>iPhone 3,1</td>
<td>June 2010</td>
</tr>
<tr>
<td>iPhone 4 - CDMA</td>
<td>A1349</td>
<td>iOS 4.0</td>
<td>N92AP</td>
<td>iPhone 3,2</td>
<td>June 2010</td>
</tr>
</tbody>
</table>

For More Information:

The most recent iPhones, the seventh generation iPhone 5C and iPhone 5S, were just released at the time of writing this book. Currently, there is no method or tool available to physically recover data from these devices. However, the file system and a logical acquisition can be obtained if the iPhone is unlocked. Acquisition methods for data extraction are available and will be discussed in Chapter 3, Data Acquisition from iOS Devices, and Chapter 4, Data Acquisition from iOS Backups.

Before examining an iPhone, it is necessary to identify the correct hardware model and the firmware version installed on the device. Knowing the iPhone details helps you to understand the criticalities and possibilities of obtaining evidence from the iPhone. For example, in many cases, the device passcode is required in order to obtain the file system or logical image. Depending on the iOS version, device model, and passcode complexity, it may be possible to obtain the device passcode using a brute force attack.

---

**For More Information:**
Understanding the Internals of iOS Devices

There are various ways to identify the hardware of a device. The easiest way to identify the hardware of a device is by observing the **model number** displayed on the back of the device. The following image shows the model number etched on the back of the casing. Apple's knowledge base articles can be helpful for this purpose. Details on identifying iPhone models can be found at [http://support.apple.com/kb/HT3939](http://support.apple.com/kb/HT3939).

![iPhone model number located on the back of the case](image)

The firmware version of an iPhone can be found by accessing the **Settings** option and then navigating to **General** | **About** | **Version**, as shown in the following screenshot. The purpose of the firmware is to enable certain features and assist with the general functioning of the device.

![The iPhone About screen, displaying firmware Version 5.1.1 (9B206)](image)

For More Information:
Alternatively, the `ideviceinfo` command-line tool available in the `libimobiledevice` software library (http://www.libimobiledevice.org/) can be used to identify the iPhone model and its iOS version. The library allows you to communicate with an iPhone even if the device is locked by a passcode. The software library was developed by Nikias Bassen (pimskeks), and it was compiled for Mac OS X by Ben Clayton (benvium).

Mac OS X can be installed in virtual machines for use on a Windows platform. To obtain the iPhone model and its iOS version information on Mac OS X 10.8, the following steps must be followed:

1. Open the terminal application.
2. From the command line, run the following command to download the `libimobiledevice` library:

   `$ git clone https://github.com/benvium/libimobiledevice-macosx.git ~/Desktop/libimobiledevice-macosx/`

   The command creates the `libimobiledevice-macosx` directory on the user's desktop and places the `libimobiledevice` command-line tools onto it.

3. Navigate to the `libimobiledevice-macosx` directory, as follows:

   `$ cd ~/Desktop/libimobiledevice-macosx/`

4. Create and edit the `.bash_profile` file using the `nano` command, as follows:

   `$ nano ~/.bash_profile`

5. Add the following two lines to the `.bash_profile` file, as follows:

   ```
   export DYLD_LIBRARY_PATH=~/Desktop/libimobiledevice-macosx/:$DYLD_LIBRARY_PATH
   PATH=${PATH}:~/Desktop/libimobiledevice-macosx/
   ```

   Press Ctrl + X, type the letter y and hit Enter to save the file.

6. Return to the terminal and run the following command:

   `$ source ~/.bash_profile`

7. Connect the iPhone to the Mac workstation using a USB cable, and run the `ideviceinfo` command with the `-s` option:

   `$ ./ideviceinfo -s`

   Output of the `ideviceinfo` command displays the iPhone identifier, internal name, and the iOS version as shown:

   ```
   BuildVersion: 9B206
   ```

For More Information:
Understanding the Internals of iOS Devices

DeviceClass: iPhone
DeviceName: iPhone 4
HardwareModel: N90AP
ProductVersion: 5.1.1
ProductionSOC: true
ProtocolVersion: 2
TelephonyCapability: true
UniqueChipID: 1937316564364
WiFiAddress: 58:1f:aa:22:d1:0a

Every release of the iPhone comes with improved or newly added features. The following tables show the specifications and features of legacy and current iPhone models:

<table>
<thead>
<tr>
<th>Specification</th>
<th>iPhone</th>
<th>iPhone 3G</th>
<th>iPhone 3GS</th>
</tr>
</thead>
<tbody>
<tr>
<td>System on chip</td>
<td>Samsung Chip</td>
<td>Samsung Chip</td>
<td>Samsung Chip</td>
</tr>
<tr>
<td></td>
<td>620 MHz Samsung 32-bit RISC ARM</td>
<td>620 MHz Samsung 32-bit RISC ARM</td>
<td>833 MHz ARM Cortex-A8</td>
</tr>
<tr>
<td>CPU</td>
<td>620 MHz Samsung 32-bit RISC ARM</td>
<td>620 MHz Samsung 32-bit RISC ARM</td>
<td>833 MHz ARM Cortex-A8</td>
</tr>
<tr>
<td>Onboard RAM</td>
<td>128 MB</td>
<td>128 MB</td>
<td>256 MB</td>
</tr>
<tr>
<td>Screen size (in inches)</td>
<td>3.5</td>
<td>3.5</td>
<td>3.5</td>
</tr>
<tr>
<td>Resolution</td>
<td>480*320</td>
<td>480*320</td>
<td>480*320</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Wi-Fi, Bluetooth 2.0, GSM</td>
<td>Wi-Fi, Bluetooth 2.0, GSM/UMTS/HSDPA, GPS</td>
<td>Wi-Fi, Bluetooth 2.1, GSM, UMTS/HSDPA, GPS</td>
</tr>
<tr>
<td>Camera (megapixel)</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Front camera</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Storage (GB)</td>
<td>4, 8, 16</td>
<td>8, 16</td>
<td>8, 16, 32</td>
</tr>
<tr>
<td>Weight (in ounces)</td>
<td>4.8</td>
<td>4.7</td>
<td>4.8</td>
</tr>
<tr>
<td>Dimensions</td>
<td>4.5 * 2.4 * 0.46</td>
<td>4.55 * 2.44 * 0.48</td>
<td>4.55 * 2.44 * 0.48</td>
</tr>
</tbody>
</table>

For More Information:
## Specification

<table>
<thead>
<tr>
<th>Specification</th>
<th>iPhone</th>
<th>iPhone 3G</th>
<th>iPhone 3GS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery life</td>
<td>8/7/6/24</td>
<td>5/7/5/24</td>
<td>5/10/5/30</td>
</tr>
<tr>
<td>Talk/video/web/audio</td>
<td>5/7/5/24</td>
<td>250</td>
<td>300</td>
</tr>
<tr>
<td>Standby time (hours)</td>
<td>250</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Colors</td>
<td>Black</td>
<td>Black, white (white not in 8 GB)</td>
<td>Black, white (white not in 8 GB)</td>
</tr>
<tr>
<td>Material</td>
<td>Aluminum, glass, and steel</td>
<td>Glass, plastic, and steel</td>
<td>Glass, plastic, and steel</td>
</tr>
<tr>
<td>Connector</td>
<td>USB 2.0 dock connector</td>
<td>USB 2.0 dock connector</td>
<td>USB 2.0 dock connector</td>
</tr>
<tr>
<td>SIM card form-factor</td>
<td>Mini SIM</td>
<td>Mini SIM</td>
<td>Mini SIM</td>
</tr>
<tr>
<td>Siri support</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

The most recent iPhone features are shown in the following table:

<table>
<thead>
<tr>
<th>Specification</th>
<th>iPhone 4</th>
<th>iPhone 4S</th>
<th>iPhone 5</th>
<th>iPhone 5C</th>
<th>iPhone 5S</th>
</tr>
</thead>
<tbody>
<tr>
<td>System on chip</td>
<td>Apple A4</td>
<td>Apple A5</td>
<td>Apple A6</td>
<td>Apple A6</td>
<td>Apple A7</td>
</tr>
<tr>
<td>CPU</td>
<td>1 GHz ARM Cortex-A8</td>
<td>800 MHz dual core ARM Cortex-A9</td>
<td>1.3 GHz dual core Apple-designed ARMv7s</td>
<td>1.3 GHz dual core Apple-designed ARMv7s</td>
<td>1.3 GHz dual core Apple-designed ARMv8-A</td>
</tr>
<tr>
<td>Onboard RAM</td>
<td>512 MB</td>
<td>512 MB</td>
<td>1 GB</td>
<td>1 GB</td>
<td>1 GB</td>
</tr>
<tr>
<td>Screen size (in inches)</td>
<td>3.5</td>
<td>3.5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Resolution</td>
<td>960*640</td>
<td>960*640</td>
<td>1136*640</td>
<td>1136*640</td>
<td>1136*640</td>
</tr>
</tbody>
</table>

For More Information:  
### Understanding the Internals of iOS Devices

<table>
<thead>
<tr>
<th>Specification</th>
<th>iPhone 4</th>
<th>iPhone 4S</th>
<th>iPhone 5</th>
<th>iPhone 5C</th>
<th>iPhone 5S</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connectivity</strong></td>
<td>Wi-Fi, Bluetooth 2.1, GSM, UMTS/HSUPA/GPS</td>
<td>Wi-Fi, Bluetooth 4, GSM, UMTS/HSUPA/GPS</td>
<td>Wi-Fi, Bluetooth 4, UMTS/HSUPA+/DC-HSDPA/DC-HSDPA+/, LTE, GSM, GPS</td>
<td>Wi-Fi, Bluetooth 4, UMTS/HSUPA+/DC-HSDPA/LTE, GSM, GPS</td>
<td>Wi-Fi, Bluetooth 4, UMTS/HSUPA+/DC-HSDPA/LTE, GSM, GPS</td>
</tr>
<tr>
<td><strong>Camera (megapixel)</strong></td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td><strong>Front camera</strong></td>
<td>VGA</td>
<td>VGA</td>
<td>720P</td>
<td>720P</td>
<td>720P</td>
</tr>
<tr>
<td><strong>Storage (GB)</strong></td>
<td>8, 16, 32</td>
<td>8, 16, 32, 64</td>
<td>16, 32, 64</td>
<td>8, 16, 32, 64</td>
<td>8, 16, 32, 64</td>
</tr>
<tr>
<td><strong>Weight (in ounces)</strong></td>
<td>4.8</td>
<td>4.9</td>
<td>3.95</td>
<td>4.7</td>
<td>4</td>
</tr>
<tr>
<td><strong>Dimensions</strong></td>
<td>4.5 * 2.31 * 0.37</td>
<td>4.5 * 2.31 * 0.37</td>
<td>4.87 * 2.31 * 0.30</td>
<td>4.98 * 2.33 * 0.353</td>
<td>4.87 * 2.31 * 0.30</td>
</tr>
<tr>
<td><strong>Battery life</strong></td>
<td>7/10/10/40</td>
<td>8/10/9/40</td>
<td>8/10/10/40</td>
<td>10/10/10/40</td>
<td>10/10/10/40</td>
</tr>
<tr>
<td><strong>Standby time</strong></td>
<td>300</td>
<td>300</td>
<td>225</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td><strong>Colors</strong></td>
<td>Black</td>
<td>Black, white</td>
<td>Black, white</td>
<td>White, pink, yellow, blue, or green</td>
<td>Silver, space gray, or gold</td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>Aluminosilicate glass and stainless steel</td>
<td>Aluminosilicate glass and stainless steel</td>
<td>Black-anodized aluminum slate metal white - silver aluminum metal</td>
<td>White, pink, yellow, blue, or green</td>
<td>White or black front with aluminum metal back</td>
</tr>
<tr>
<td><strong>Connector</strong></td>
<td>USB 2.0 dock connector</td>
<td>USB 2.0 dock connector</td>
<td>Lightning connector</td>
<td>Lightning connector</td>
<td>Lightning connector</td>
</tr>
<tr>
<td><strong>SIM card form-factor</strong></td>
<td>Micro SIM</td>
<td>Micro SIM</td>
<td>Nano-SIM</td>
<td>Nano-SIM</td>
<td>Nano-SIM</td>
</tr>
<tr>
<td><strong>Siri support</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For More Information:

One of the major changes in the iPhone 5, iPhone 5C, and iPhone 5S is the USB dock connector, which is used to charge and synchronize the device with the computer. Devices prior to the iPhone 5 use a 30-pin USB dock connector, whereas the newer iPhones use an eight-pin lightning connector.

**iPhone hardware**

The iPhone is a collection of modules, chips, and electronic components from different manufacturers. Due to the complexities of the iPhone, the list of hardware components is extensive. A detailed list of iPhone hardware components is defined at https://viaforensics.com/resources/white-papers/iphone-forensics/overview.

The following images show the internals of the iPhone 5S. The images were taken after dismantling the iPhone 5S. Internal images for all iPhones can be found in the teardown section from http://www.ifixit.com/Device/iPhone.

The iPhone 5S teardown image—side one (included with kind permission from TechInsights)

For More Information:  
And the following is the image showing the back of the iPhone 5S:

![iPhone 5S teardown image - side two](image)

The iPhone 5S teardown image—side two (included with kind permission from TechInsights)

## iPad models

The Apple iPhone changed the way cell phones are produced and used. Similarly, the iPad, a version of the tablet computer introduced in January 2010, squashed the sales of notebooks. With the iPad, individuals can shoot video, take photos, play music, read books, browse the Internet, and do much more. Various iPad models exist now with different features and storage capabilities. The following table lists all the iPad models and their initial iOS versions. Details on identifying iPad models can be found at [http://support.apple.com/kb/ht5452](http://support.apple.com/kb/ht5452).

<table>
<thead>
<tr>
<th>Device</th>
<th>Model</th>
<th>Initial OS</th>
<th>Internal name</th>
<th>Identifier</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPad - Wi-Fi</td>
<td>A1219</td>
<td>iOS 3.2</td>
<td>K48AP</td>
<td>iPad 1,1</td>
<td>January 2010</td>
</tr>
<tr>
<td>iPad - 3G</td>
<td>A1337</td>
<td></td>
<td></td>
<td>iPad 1,1</td>
<td></td>
</tr>
<tr>
<td>iPad 2 - Wi-Fi</td>
<td>A1395</td>
<td>iOS 4.3</td>
<td>K93AP</td>
<td>iPad 2,1</td>
<td>March 2011</td>
</tr>
<tr>
<td>iPad 2 - GSM</td>
<td>A1396</td>
<td></td>
<td>K94AP</td>
<td>iPad 2,2</td>
<td></td>
</tr>
<tr>
<td>iPad 2 - CDMA</td>
<td>A1397</td>
<td></td>
<td>K95AP</td>
<td>iPad 2,3</td>
<td></td>
</tr>
</tbody>
</table>

For More Information:
<table>
<thead>
<tr>
<th>Device</th>
<th>Model</th>
<th>Initial OS</th>
<th>Internal name</th>
<th>Identifier</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPad 2 - Wi-Fi rev</td>
<td>A1395</td>
<td></td>
<td>K93AAP</td>
<td>iPad 2,4</td>
<td></td>
</tr>
<tr>
<td>iPad 3 - Wi-Fi</td>
<td>A1416</td>
<td></td>
<td>J1AP</td>
<td>iPad 3,1</td>
<td></td>
</tr>
<tr>
<td>iPad 3 - Wi-Fi + Cellular Verizon</td>
<td>A1403</td>
<td>iOS 5.1</td>
<td>J2AP</td>
<td>iPad 3,2</td>
<td>March 2012</td>
</tr>
<tr>
<td>iPad 3 - Wi-Fi + Cellular AT&amp;T</td>
<td>A1430</td>
<td></td>
<td>J2AAP</td>
<td>iPad 3,3</td>
<td></td>
</tr>
<tr>
<td>iPad 4 - Wi-Fi</td>
<td>A1458</td>
<td>iOS 6.0</td>
<td>P101AP</td>
<td>iPad 3,4</td>
<td></td>
</tr>
<tr>
<td>iPad 4 - Wi-Fi + Cellular AT&amp;T</td>
<td>A1459</td>
<td></td>
<td>P102AP</td>
<td>iPad 3,5</td>
<td></td>
</tr>
<tr>
<td>iPad 4 - Wi-Fi + Cellular Verizon</td>
<td>A1460</td>
<td>iOS 6.0.1</td>
<td>P103AP</td>
<td>iPad 3,6</td>
<td></td>
</tr>
<tr>
<td>iPad mini - Wi-Fi</td>
<td>A1432</td>
<td>iOS 6.0</td>
<td>P105AP</td>
<td>iPad 2,5</td>
<td>October 2012</td>
</tr>
<tr>
<td>iPad mini - Wi-Fi + Cellular AT&amp;T</td>
<td>A1454</td>
<td></td>
<td>P106AP</td>
<td>iPad 2,6</td>
<td></td>
</tr>
<tr>
<td>iPad mini - Wi-Fi + Cellular Verizon and Sprint</td>
<td>A1455</td>
<td>iOS 6.0.1</td>
<td>P107AP</td>
<td>iPad 2,7</td>
<td></td>
</tr>
<tr>
<td>iPad Air - Wi-Fi</td>
<td>A1474</td>
<td>iOS 7.0.3</td>
<td>J71AP</td>
<td>iPad 4,1</td>
<td>November 2013</td>
</tr>
<tr>
<td>iPad Air - Wi-Fi + Cellular</td>
<td>A1475</td>
<td>iOS 7.0.3</td>
<td>J72AP</td>
<td>iPad 4,2</td>
<td></td>
</tr>
</tbody>
</table>

For More Information:  
Every release of the iPad comes with improved or newly added features. The following table shows the specifications and features of legacy and current iPad Wi-Fi models:

<table>
<thead>
<tr>
<th>Specification</th>
<th>iPad</th>
<th>iPad 2</th>
<th>iPad 3</th>
<th>iPad 4</th>
<th>iPad Mini</th>
<th>iPad Air</th>
</tr>
</thead>
<tbody>
<tr>
<td>System on chip</td>
<td>Apple A4</td>
<td>Apple A5</td>
<td>Apple A5X</td>
<td>Apple A6X</td>
<td>Apple A5</td>
<td>Apple A7</td>
</tr>
<tr>
<td>CPU</td>
<td>1GHz dual core Samsung-Intensity</td>
<td>1 GHz dual core ARM Cortex-A9</td>
<td>1 GHz dual core ARM Cortex-A9</td>
<td>1.4 GHz dual core Apple Swift</td>
<td>1 GHz dual core ARM Cortex-A9</td>
<td>1.4 GHz dual core ARMv8-A</td>
</tr>
<tr>
<td>Onboard RAM</td>
<td>256 MB</td>
<td>512 MB</td>
<td>1 GB</td>
<td>1 GB</td>
<td>512 MB</td>
<td>1 GB</td>
</tr>
<tr>
<td>Screen size (in inches)</td>
<td>9.7</td>
<td>9.7</td>
<td>9.7</td>
<td>9.7</td>
<td>7.9</td>
<td>9.7</td>
</tr>
<tr>
<td>Resolution</td>
<td>1024*768</td>
<td>1024*768</td>
<td>2048*1536</td>
<td>2048*1536</td>
<td>1024*768</td>
<td>2048*1536</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Wi-Fi, Bluetooth 2.1</td>
<td>Wi-Fi, Bluetooth 2.1</td>
<td>Wi-Fi, Bluetooth 4</td>
<td>Wi-Fi, Bluetooth 4</td>
<td>Wi-Fi, Bluetooth 4</td>
<td>Wi-Fi, Bluetooth 4</td>
</tr>
<tr>
<td>Camera (megapixel)</td>
<td>N/A</td>
<td>0.7</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Front camera</td>
<td>N/A</td>
<td>0.3 MP</td>
<td>0.3 MP</td>
<td>1.2 MP</td>
<td>1.2 MP</td>
<td>1.2 MP</td>
</tr>
<tr>
<td>Storage (GB)</td>
<td>16, 32, 64</td>
<td>16, 32, 64</td>
<td>16, 32, 64</td>
<td>16, 32, 64, 128</td>
<td>16, 32, 64</td>
<td>16, 32, 64, 128</td>
</tr>
<tr>
<td>Weight (in ounces)</td>
<td>24</td>
<td>21.6</td>
<td>22.9</td>
<td>22.9</td>
<td>10.8</td>
<td>16</td>
</tr>
<tr>
<td>Dimensions</td>
<td>9.56 * 7.47 * 0.5</td>
<td>9.5 * 7.31 * 0.34</td>
<td>9.5 * 7.31 * 0.37</td>
<td>9.5 * 7.31 * 0.37</td>
<td>7.87 * 5.31 * 0.28</td>
<td>9.4 * 6.6 * 0.29</td>
</tr>
<tr>
<td>Battery life</td>
<td>10/10/140</td>
<td>10/10/140</td>
<td>10/10/140</td>
<td>10/10/140</td>
<td>10/10/140</td>
<td>10/10/140</td>
</tr>
<tr>
<td>Video/web/audio</td>
<td>1 month</td>
<td>1 month</td>
<td>1 month</td>
<td>1 month</td>
<td>1 month</td>
<td>1 month</td>
</tr>
<tr>
<td>Standby time (hours)</td>
<td>USB 2.0 dock connector</td>
<td>USB 2.0 dock connector</td>
<td>USB 2.0 dock connector</td>
<td>Lightning connector</td>
<td>Lightning connector</td>
<td>Lightning connector</td>
</tr>
<tr>
<td>Connector</td>
<td>USB 2.0 dock connector</td>
<td>USB 2.0 dock connector</td>
<td>USB 2.0 dock connector</td>
<td>Lightning connector</td>
<td>Lightning connector</td>
<td>Lightning connector</td>
</tr>
</tbody>
</table>

For More Information:
iPod hardware

One of the key successes of Apple iOS devices is the proper selection of its hardware components. Just like the iPhone, the iPad is also a collection of modules, chips, and electronic components from different manufacturers. Internal images for all iPads can be found in the teardown section of http://www.ifixit.com/Device/iPad.

The following images show the internals of the iPad 3. The images were taken after dismantling the iPad 3 cellular model and were obtained from http://www.chipworks.com/.

The iPad 3 cellular model teardown image—side one (included with kind permission from Chipworks)
File system
To better understand the forensic process of an iPhone, it is good to know about the file system that is used. The file system used in the iPhone and other Apple iOS devices is HFSX, a variation of HFS Plus with one major difference. HFSX is case sensitive whereas HFS Plus is case insensitive. Other differences will be discussed later in this chapter. OS X uses HFS Plus by default and iOS uses HFSX.

The HFS Plus file system
In 1996, Apple developed a new file system, Hierarchical File System (HFS), to accommodate the storage of large datasets. In an HFS file system the storage medium is represented as volumes. HFS volumes are divided into logical blocks of 512 bytes. The logical blocks are numbered from first to last on a given volume and will remain static with the same size as physical blocks, that is, 512 bytes. These logical blocks are grouped together into allocation blocks, which are used by the HFS file system to track data in a more efficient way. HFS uses a 16-bit value to address allocation blocks, which limits the number of allocation blocks to 65,535. To overcome the inefficient allocations of disk space and some of the limitations of HFS, Apple introduced the HFS Plus file system (http://dubeiko.com/development/FileSystem/HFSPLUS/tn1150.html).

For More Information:
The HFS Plus file system was designed to support larger file sizes. HFS volumes are divided into sectors that are usually 512 bytes in size. These sectors are grouped together into allocation blocks. The number of allocation blocks depends on the total size of the volume. HFS Plus uses block addresses of 32 bits to address allocation blocks. HFS Plus uses journaling by default. Journaling is the process of logging every transaction to the disk, which helps in preventing file system corruption. The key characteristics of the HFS Plus file system are: efficient use of disk space, unicode support for filenames, support for name forks, file compression, journaling, dynamic resizing, dynamic defragmentation, and an ability to boot on operating systems other than Mac OS.

**The HFS Plus volume**

The HFS Plus volume contains a number of internal structures to manage the organization of data. These structures include a header, alternate header, and five special files: an allocation file, an Extents Overflow file, a Catalog file, an Attributes file, and a Startup file. Among the five files, three files, the Extents Overflow file, the Catalog file, and the Attribute file, use a B-tree structure, a data structure that allows data to be efficiently searched, viewed, modified, or removed. The HFS Plus volume structure is shown in the following figure:

---

For More Information:  
The volume structure is described as follows:

- The first 1,024 bytes are reserved for boot load information.
- **Volume Header**: This stores volume information, such as the size of allocation blocks, a timestamp of when the volume was created, and metadata about each of the five special files.
- **Allocation File**: This file is used to track which allocation blocks are in use by the system. The file format consists of one bit for every allocation block. If the bit is set, the block is in use. If it is not set, the block is free.
- **Extents Overflow File**: This file records the allocation blocks that are allocated when the file size exceeds eight blocks, which helps in locating the actual data when referred. Bad blocks are also recorded in the file.
- **Catalog File**: This file contains information about the hierarchy of files and folders, which is used to locate any file and folder within the volume.
- **Attribute File**: This file contains inline data attribute records, fork data attribute records, and extension attribute records.
- **Startup File**: This file holds the information needed to assist in booting a system that does not have HFS Plus support.
- **Alternate Volume Header**: This is a backup of the volume header, and it is primarily used for disk repair.
- The last 512 bytes are reserved for use by Apple, and it is used during the manufacturing process.

**Disk layout**

By default, the file system is configured as two logical disk partitions: system (root or firmware) partition and user data partition.

The system partition contains the OS and all of the preloaded applications used with the iPhone. The system partition is mounted as read-only unless an OS upgrade is performed or the device is jailbroken. The partition is updated only when a firmware upgrade is performed on the device. During this process, the entire partition is formatted by iTunes without affecting any of the user data. The system partition takes only a small portion of storage space, normally between 0.9 GB and 2.7 GB, depending on the size of the NAND drive. As the system partition was designed to remain in factory state for the entire life of the iPhone, there is typically little useful evidentiary information that can be obtained from it. If the iOS device was jailbroken, files containing information regarding the jailbreak may be resident on the system partition. Jailbreaking an iOS device allows the user root access to the device and voids the manufacturer warranty. Jailbreaking will be discussed later in this chapter.

---

**For More Information:**

The user data partition contains all user-created data ranging from music to contacts. The user data partition occupies most of the NAND memory and is mounted at `/private/var` on the device. Most of the evidentiary information can be found in this partition. During a physical acquisition, both the user data and system partitions can be captured and saved as a `.dmg` or `.img` file. These raw image files can be mounted as read-only for forensic analysis, which is covered in detail in Chapter 3, *Data Acquisition from iOS Devices*. Even on non-jailbroken iOS devices, it is recommended to acquire both the system and user data partitions to ensure all data is obtained for examination.

To view the mounted partitions on the iPhone, connect a jailbroken iPhone to a workstation over SSH, and run the `mount` command. For this example, iPhone 4 with 5.1.1 is used.

The `mount` command shows that the system partition is mounted on `/ (root)`, and the user data partition is mounted on `/private/var`, as shown in the following command lines. Both partitions show HFS as the file system, and the user data partition even shows that journaling is enabled.

```
iPhone4:~ root# mount
/dev/disk0s1s1 on / (hfs, local, journaled, noatime)
devfs on /dev (devfs, local, nobrowse)
/dev/disk0s1s2 on /private/var (hfs, local, journaled, noatime, protect)
```

To view the raw disk images on the iPhone, connect a jailbroken iPhone to a workstation over SSH, and run the `ls -lh rdisk*` command. `rdisk0` is the entire file system and `rdisk0s1` is the firmware partition. `rdisk0s1s1` is the root file system and `rdisk0s1s2` is the user file system, as shown in the following command lines:

```
iPhone4:/dev root# ls -lh rdisk*
crw-r----- 1 root operator 14, 0 Oct 10 04:28 rdisk0
crw-r----- 1 root operator 14, 1 Oct 10 04:28 rdisk0s1
crw-r----- 1 root operator 14, 2 Oct 10 04:28 rdisk0s1s1
crw-r----- 1 root operator 14, 3 Oct 10 04:28 rdisk0s1s2
```

**iPhone operating system**

iOS is Apple's most advanced and feature-rich proprietary mobile operating system. It was released with the first generation of the iPhone. When introduced, it was named *iPhone OS*, and later it was renamed to *iOS* to reflect the unified nature of the operating system that powers all Apple iOS devices, such as the iPhone, iPod Touch, iPad, and Apple TV. iOS is derived from core OS X technologies and streamlined to be compact and efficient for mobile devices.

For More Information:

It utilizes a multitouch interface where simple gestures are used to operate and control the device, such as swiping your finger across the screen to move to the successive page or pinching your fingers to zoom. In simple terms, iOS assists with the general functioning of the device. iOS is really Mac OS X with some significant differences:

- The architecture for which the kernel and binaries are compiled is ARM-based rather than Intel x86_64
- The OS X kernel is open source, whereas the iOS kernel remains closed
- Memory management is much tighter
- The system is hardened and does not allow access to the underlying APIs

iOS history

iOS, like any other operating system, has gone through multiple updates since its release. Apple occasionally releases newer versions to enable new features, to support latest hardware, and to fix bugs. The latest version of iOS at the time of this writing is iOS 7.0.3. Though Apple sticks with a numeric approach for new iOS builds, all iOS versions have code names that are private to Apple. The following sections describe the history of iOS development.

1.x – the first iPhone

iPhone OS 1.x was the first release of Apple's touch-centric mobile operating system. On its initial release, Apple stated that the iPhone uses a version of the desktop operating system, OS X. Later it was named iPhone OS. The original build was known as Alpine, but the final released version was Heavenly.

2.x – App Store and 3G

iPhone OS 2.0 (known as BigBear) was released along with iPhone 3G. Features required for corporate needs such as VPN and Microsoft Exchange were introduced with this release. The big addition to the OS with this release was the App Store, a marketplace for the third-party applications that could run on the iPhone. Apple also released the iPhone Software Development Kit (SDK) to assist developers in creating applications on the App Store for free or for purchase. Global Positioning System (GPS) was also added to the iPhone with this release.

For More Information:
3.x – the first iPad

iPhone OS 3.0 (known as Kirkwood) became available with the release of iPhone 3GS. The iOS release brought the copy/paste feature, spotlight searches, and push notifications for third-party applications, and many other enhancements to the built-in applications. Multitasking was introduced, but it was limited to a selection of the applications Apple included on the device. The first iPad was introduced with iPhone OS 3.2 (known as Wildcat) and later updated to 3.2.2, a version specifically made for the iPad.

4.x – Game Center and multitasking

iOS 4.0 (known as Apex) was the first major release after renaming the iPhone OS to iOS. This release brought over 100 new features, such as FaceTime, iBooks, voice control, and 1,500 new APIs to the developers. Starting with this release, multitasking was extended to third-party iOS applications. Apple also released Game Center, an online multiplayer social gaming network along with this release.

5.x – Siri and iCloud

iOS 5.0 (known as Telluride) was released with iPhone 4S. iOS 5 with iPhone 4S introduced Apple's natural language-based voice control, Siri—a virtual assistant. This update brought many new features, such as notification center, iMessages, Newsstand, Twitter integration, the Reminders application, and over the air (OTA) software updates. The biggest addition to the release was the iCloud, Apple's cloud-based service that allows users to synchronize their contacts, calendar, pictures, and much more to the cloud.

6.x – Apple Maps

iOS 6.0 (known as Sundance) was released in June 2012 with the release of iPhone 5. With iOS 6, the old, Google-powered Maps application was removed, and an all-new Apple Maps with data supplied by TomTom was added. The YouTube application was also removed in this update. iOS 6 brought many new features, such as Facebook integration, FaceTime over cellular network, Passbook, and many enhancements to the built-in applications. Better privacy controls were added with this release.

For More Information:
7.x – the iPhone 5S and beyond

iOS 7.0 (known as Innsbruck) was released in September 2013 with the release of iPhone 5S. The biggest change in iOS 7 and the most important was the system-wide redesign. With this release, Apple took the interface experience from static to dynamic. A ton of new features were introduced, such as control center, AirDrop, iTunes Radio, FaceTime audio, automatic updates for applications, activation lock, and many more. With iPhone 5S, Apple’s Touch ID fingerprint identity sensor, a biometric authentication technology, was introduced.

All the iOS versions are not supported by all the iOS devices. Each iOS version is compatible only with a few devices, as shown in the following iOS compatibility matrix. This table was created using http://iossupportmatrix.com/. The blocks in green signify that an iOS version was supported for that device. If a version is listed, it is the earliest version supported for that device. The blocks in red mean no support for that device, and the blocks in blue are still iOS versions supported by Apple.

<table>
<thead>
<tr>
<th></th>
<th>iPhone OS 1.0</th>
<th>iPhone SDK 2.0</th>
<th>iPhone SDK 3.0</th>
<th>iPhone SDK 4.0</th>
<th>iOS 5</th>
<th>iOS 6</th>
<th>iOS 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone</td>
<td>1.0</td>
<td>3.1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPod Touch</td>
<td>1.1</td>
<td>3.1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPhone 3G</td>
<td>2.0</td>
<td>4.2.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPod Touch (2nd Gen)</td>
<td>2.1.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPhone 3GS</td>
<td>3.0</td>
<td></td>
<td>6.1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPod Touch (3rd Gen)</td>
<td>3.1.1</td>
<td></td>
<td></td>
<td>5.1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPad (1st Gen)</td>
<td>3.2</td>
<td></td>
<td>4.3.5</td>
<td>5.1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPhone 4</td>
<td></td>
<td>4.0 (GSM)/4.2.6 (CDMA)</td>
<td></td>
<td>7.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPod Touch (4th Gen)</td>
<td></td>
<td>4.2.1</td>
<td></td>
<td>6.1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPad2</td>
<td></td>
<td>4.3.5</td>
<td></td>
<td>7.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPhone 4S</td>
<td></td>
<td></td>
<td>5.0</td>
<td>7.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPad</td>
<td></td>
<td></td>
<td>5.1</td>
<td>7.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPod Touch (5th Gen)</td>
<td></td>
<td></td>
<td></td>
<td>6.0</td>
<td>7.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPad Mini</td>
<td></td>
<td></td>
<td>6.0</td>
<td>7.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPhone 5</td>
<td></td>
<td></td>
<td>6.0</td>
<td>7.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPhone 5C</td>
<td></td>
<td></td>
<td></td>
<td>7.0</td>
<td>7.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPhone 5S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.0.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The OS compatibility matrix
The iOS architecture

iOS acts as an intermediary between the underlying hardware components and the applications that appear on the screen. The applications do not talk to the underlying hardware directly. Instead, they communicate through a well-defined system interface that protects the applications from hardware changes. This abstraction makes it easy to build applications that work on devices with different hardware capability.

The iOS architecture consists of four layers: the Cocoa Touch layer, Media layer, Core Services layer, and Core OS layer, as shown in the following figure. Each layer consists of several frameworks that would help to build an application.

The Cocoa Touch layer

The Cocoa Touch layer contains the key frameworks required to develop the visual interface for iOS applications. Frameworks in this layer provide the basic application infrastructure and support key technologies, such as multitasking and touch-based input, and many high-level system services.

The Media layer

The Media layer provides the graphics and audio and video frameworks to create the best multimedia experience available on a mobile device. The technologies in this layer help developers to build applications that look and sound great.

The Core Services layer

This Core Services layer provides the fundamental system services that are required for the applications. All these services are not used by the developers though many parts of the system are built on top of them. The layer contains the technologies to support features such as location, iCloud, and social media.
The Core OS layer
The Core OS layer is the base layer and sits directly on top of the device hardware. This layer deals with low-level functionalities and provides services such as networking (BSD sockets), memory management, threading (POSIX threads), file system handling, external accessories access, and inter-process communication.

iOS security
iOS was designed with security at its core. At the highest level, the iOS security architecture appears as shown in the following figure:

The iOS security architecture

For More Information:
Apple iOS devices such as iPhone, iPad, and iPod Touch are designed with layers of security. Low-level hardware features safeguard from malware attacks and the high-level OS features prevent unauthorized use. A brief overview of the iOS security features are provided in the following sections.

**Passcode**
Passcodes restrict unauthorized access to the device. Once a passcode is set, each time you turn on or wake up the device, it will ask for the passcode to access the device. iPhone supports simple as well as complex passcodes. iPhone 5S also supports touch ID fingerprints as a passcode.

**Code signing**
Code signing prevents users from downloading and installing unauthorized applications on the device. Apple says "Code Signing is the process by which your compiled iOS application is sealed and identified as yours. Also, iOS devices won't run an application or load a library unless it is signed by a trusted party. To ensure that all apps come from a known and approved source and have not been tampered with, iOS requires that all executable code be signed using an Apple-issued certificate."

**Sandboxing**
Sandboxing mitigates the post-code-execution exploitation by placing the application into a tightly restricted area. Applications installed on the iOS device are sandboxed, and one application cannot access the data stored by the other application.

**Encryption**
On iOS devices, the entire file system is encrypted with a file system key, which is computed from the device's unique hardware key.

**Data protection**
Data protection is designed to protect data at rest and to make offline attacks difficult. It allows applications to leverage the user's device passcode in concert with the device hardware encryption to generate a strong encryption key. Later, the strong encryption key is used to encrypt the data stored on the disk. This key prevents data from being accessed when the device is locked, ensuring that critical information is secured even if the device is compromised.

For More Information:
Address Space Layout Randomization
Address Space Layout Randomization (ASLR) is an exploit mitigation technique introduced with iOS 4.3. ASLR randomizes the application objects' location in the memory, making it difficult to exploit the memory corruption vulnerabilities.

Privilege separation
iOS runs with the principle of least privileges. It contains two user roles: root and mobile. The most important processes in the system run with root user privileges. All other applications that the user has direct access to, such as the browser and third-party applications, run with mobile user privileges.

Stack smashing protection
Stack smashing protection is an exploit mitigation technique. It protects against buffer overflow attacks by placing a random and known value (called stack canary) between a buffer and control data on the stack.

Data execution prevention
Data execution prevention (DEP) is an exploit mitigation technique mechanism in which a processor can distinguish the portions of memory that are executable code from data.

Data wipe
iOS provides an option Erase All Content and Settings to wipe the data on the iPhone. This type of data wipe erases user settings and information by removing the encryption keys that protects the data. As the encryption keys are erased from the device, it is not possible to recover the deleted data in forensic investigations. Other wiping methods are available that overwrite the data in the device memory. More information on wiping can be found at http://support.apple.com/kb/ht2110.

Activation Lock
Activation Lock, introduced with iOS 7, is a theft deterrent that works by leveraging Find My iPhone. When Find My iPhone is enabled, it enables the Activation Lock, and your Apple ID and password will be required to turn off Find My iPhone, to erase your device, and to reactive your device.
App Store
The App Store is an application distribution platform for iOS, developed and maintained by Apple. It is a centralized online store where users can browse and download both free and paid apps. These apps expand the functionality of a mobile device. As of December 2013, there are more than 1 million applications in the App Store, and users have downloaded them over 60 billion times. Apps available in the App Store are generally written by third-party developers. Developers use XCode and the iPhone SDK to develop iOS applications. Later, they submit the app to Apple for approval. Apple follows an extensive review process to check the app against the company guidelines. If Apple approves the app, it is published to the App Store where users can download or buy it. The strict review process makes the App Store less prone to malware. Currently, users can access the App Store via iTunes and also from their iOS devices.

Jailbreaking
Jailbreaking is the process of removing limitations imposed by Apple's mobile operating system through the use of software and hardware exploits. Jailbreaking permits unsigned code to run and gain root access on the operating system. The most common reason for jailbreaking is to expand the limited feature set imposed by Apple's App Store and to install unapproved apps. Many publicly available jailbreaking tools add an unofficial application installer to the device, such as Cydia, which allows users to install many third-party applications, tools, tweaks, and apps from an online file repository. The software downloaded from Cydia opens up endless possibilities on a device that a non-jailbroken device would never be able to do. The most popular jailbreaking tools are redsn0w, sn0wbreeze, evasi0n, Absinthe, seas0npass, and so on. Not all the iOS versions are jailbreakable. The website http://www.guidermyjailbreak.com/choose-iphone-to-jailbreak/ can be helpful to find out whether a particular iOS version is jailbreakable or not and with which method. In October 2012, The U.S. Copyright Office declared that jailbreaking the iPad is illegal, while jailbreaking the iPhone is deemed legal. The governing law is reviewed every three years.

For More Information:
Summary

The first step in a forensic examination of an iOS device should be identifying the device model. The model of an iOS device can be used to help the examiner develop an understanding of the underlying components and capabilities of the device, which can be used to drive the methods for acquisition and examination. Legacy iOS devices should not be disregarded because they may surface as part of an investigation. Examiners must be aware of all iOS devices as old devices are sometimes still in use and may be tied to a criminal investigation. The next chapter will provide tips and techniques for acquiring data from the iOS devices discussed in this chapter.
Where to buy this book

You can buy Practical Mobile Forensics from the Packt Publishing website:

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.