**Why did I write this book?**

Two years ago, I changed my job from network equipment work to mobile development. It was the time that mobile development was booming in China. Many startups had sprung up and social networking Apps were very popular with investors. As long as you had a good idea, you could get venture capital at scale of millions, and high salary recruitment dazzles everyone.

At that time, I had already developed some difficult enterprise Apps and I wanted to try some cooler techniques rather than developing social Apps. I thought social Apps were too easy for me. By chance, I joined the company Security Manager, built the iOS team from scratch, and took the responsibility for developing iOS Apps, including App Store and Cydia versions.

In fact, the foundation of jailbreak development is iOS reverse engineering. However, I didn't have too much experience at that time. I was totally a newbie in this area. Fortunately, I could search and learn knowledge on Google. And for iOS developers, jailbreak development and reverse engineering were not completely separate. Although the information shared on the Internet was fragmented and sometimes duplicated, they could still be organized into a complete knowledge map as long as you paid much attention.

However, studying alone makes people feel lonely, especially when you encounter a problem that no one else has encountered. Every time I had to solve problems by myself, I felt that it would be very happy if there were some skillful people that I could communicate with. Although I could email my questions to those experts like Ryan Petrich, I thought it might be some disturbance for them if my questions were very easy. So I always tried to dig into the problems and solve it by myself before I decided to ask them questions.

This embarrassing period lasted for over half a year and it ended when I met another author of this book, snakeninny, in 2012. At that time, he was a master student who faced the pressure of graduation. However, he didn't write his master thesis. Instead, he focused on the underlying iOS research and made big progress. I once asked him why not choose to develop iOS Apps since there were already lots of people engaging in it and had made large amount of money. He answered to me that his ambition was to become the leading developer in the world.

Most of time we solved problems independently. Although we just occasionally discussed questions with each other on the Internet, we still made some valuable collaborations. Before we started to write this book, we once cracked MOMO by reverse engineering and made a tweak that could show position of girls on the map. Of course, we were good developers and we submitted this bug to MOMO and they soon fixed it. This time, we cooperate again, summarize our knowledge into this book and present it to you.

During these years of research on jailbreak development and reverse engineering, the biggest payoff for me is that when I look at an iOS App, I always try to analyze it from underlying architecture and its performance. Both can directly reflect the skill level of its development team. Not only can these experiences be applied to jailbreak development, but also they are suitable for iOS App development. Of course, we must admit there are both positive and negative impacts on reverse engineering. However, we cannot deny the necessity of this area even if Apple doesn't advocate jailbreak development. If we blindly believe that the security issues exposed in this book don’t actually exist, we’re just deceiving ourselves.

Every experienced developer understands that the more knowledge you know, the more likely you have to deal with underlying technologies. For example, what aspects does the Sandbox protection mechanism apply to? Is it a pity that we only study the mechanism of runtime theoretically?

In the field of Android development, the underlying technologies are open source. However, for iOS, only the tip of the iceberg has been exposed. Although there are some books related to iOS security, such as ‘Hacking and Securing iOS Applications’, ’iOS Hacker’s Handbook’, they are too hard for most developers to understand. Even those who have some experience, like us, have difficulties in reading these books.

Since books mentioned above are too hard for most people, why not write a book consists of more details and more practical examples? So concepts, tools, theories and practices organize the contents of this book comprehensively and systematically. We illustrate our experience and knowledge from easy to hard accompanying with figures and examples, helping readers explore the internals of Apps step by step. We do not try to analyze code snippets in depth like other tech blogs. Also, we don't want to puzzle ourselves with how many similar solutions can we use to fix the same problem. What we want to do is to provide readers with a complete system of knowledge and a methodology of iOS reverse engineering. We believe that readers will gain a lot from this book.

Recently, more and more programming experts are joining the jailbreak development community. Although they keep low profile, their works, such as jailbreak tools, App assistants and Cydia tweaks, have great influence on the development of iOS. Their technique level is far beyond mine. But I'm more eager to share knowledge in the hope of helping others.

**Who are our target readers?**

People of the following kinds may find this book useful.

1. iOS enthusiasts.

2. Senior iOS developers, who have good command of App development and have the desire to understand iOS better.

3. Architects. During the process of reverse engineering, they can learn architectures of those excellent Apps so that they can improve their ability of architecture design.

4. Reverse engineers in other systems who’re also interested in iOS.

How to read this book?

There are four parts in this book. They are concepts, tools, theories and practices, respectively. The first three parts will introduce the background, knowledge and its associated tools as well as theories. The fourth part will consists of four examples so that readers will have a deeper understanding of previous knowledge in a practical way.

If the reader doesn’t have experience in iOS reverse engineering, we recommend you to start from the first part rather than start from the fourth part directly. Although it can be very cool visually, hacking is tasteless if you don't know how everything is working underneath.

**Errata and Support**

Due to our limited skills and writing schedule, it is inevitable that there are some errors or inaccuracies in the book. We plea for your correction and criticism. Also, readers can visit our official forum (http://bbs.iosre.com) and you will find iOS reverse engineers all over the world on it. Your questions will definitely get satisfied answers.

Of course, if you have any good idea or suggestion, you can get in touch with us via the forum too. We are looking forward to hearing from you and your feedback.

**Acknowledgements**

In the first place, I want to say thank you to evad3rs、PanguTeam、TaiG、saurik and other top teams and experts.

Also thanks to Dustin Howett. His Theos is a powerful tool that helped me to step into iOS reverse engineering.

Thanks to Security Manager for providing me with such nice atmosphere for studying reverse engineering. Although I have left this company, I do wish it could grow better.

Thanks to everyone who offers helps to me. Thanks for your support and encouragement.

This book is dedicated to my dearest family, and many friends who love iOS development.

Sam Wu