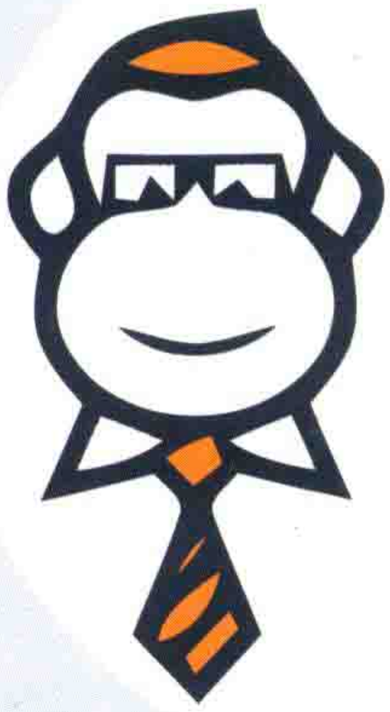
Broadview®
www.broadview.com.cn

安全技术大系



iOS应用逆向与安全

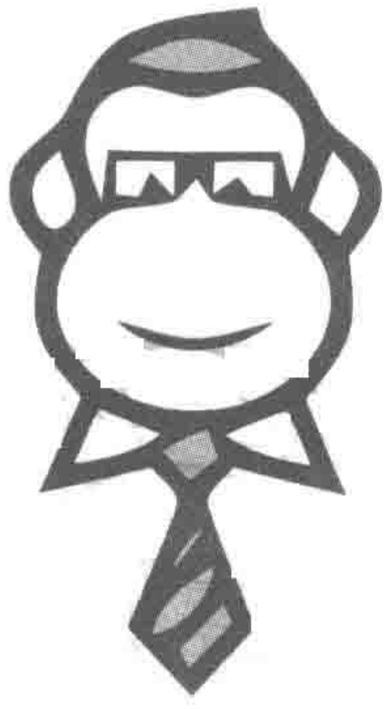
刘培庆 / 著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

安全技术大系

刘培庆 / 著

iOS应用逆向与安全



電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书从正向开发、工具的使用、逆向实战及安全保护等方面，对 iOS 应用的逆向与安全进行了讲解。本书内容包括基本概念、逆向环境的准备、常用逆向分析工具、类的结构、App 签名、Mach-O 文件格式、hook 原理等，并通过在越狱平台和非越狱平台上的逆向分析实例，带领读者学习逆向分析的思路和方法。在应用安全及保护方面，本书内容涉及网络传输、安全检测、代码混淆等。

本书适合高校计算机相关专业的学生、iOS 开发工程师、逆向工程师、越狱开发工程师、iOS 安全工程师及应用安全审计人员阅读参考。

图书在版编目（CIP）数据

iOS 应用逆向与安全 / 刘培庆著. —北京：电子工业出版社，2018.6
（安全技术大系）
ISBN 978-7-121-34099-4

Ⅰ. ①i… Ⅱ. ①刘… Ⅲ. ①移动终端—应用程序—程序设计—安全技术 Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字（2018）第 081246 号

策划编辑：潘　昕
责任编辑：潘　昕
印　　刷：三河市君旺印务有限公司
装　　订：三河市君旺印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787 × 980　1/16　　印张：25.25　　字数：535 千字
版　　次：2018 年 6 月第 1 版
印　　次：2018 年 8 月第 3 次印刷
定　　价：85.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）51260888-819，faq@phei.com.cn。

推荐序

第一次和刘培庆见面是在 2017 年，西湖大道的碧桃小馆。

四月的杭州已是初夏，只穿一件单衣的我，后背仍被汗水浸湿。昏暗的餐厅里，刘培庆坐在角落，一件牛仔夹克扣到顶，看上去又闷又热。我眉头一皱，心生不悦：这人真傻。

高温败了胃口，团购的双人餐，俩大老爷们儿竟然没吃完。结账走人，我们在涌金门一带的巷弄里穿行消食，有一搭没一搭地聊着。刘培庆说，他放弃了晋升的机会，跳到了另一个更辛苦的部门，是因为新部门的业务更吸引他。放弃更舒适的环境和更高的薪水去追求兴趣？兴趣值几个钱？这人真傻。

是夜一别，我们各自忙于生活，很久没有再会。五月某天，微信上突然弹出刘培庆的名字，“狗神，你那本书有没有打算再出一版，帮你更新一下”，然后向我介绍了他的更新计划，原来是在《iOS 应用逆向工程》的基础上更深入了。我觉得他的内容定位有些超出原书目标读者的水平，于是建议他当作技术博客发出来，算是委婉的回绝。你的原创内容，放在我的书里，如果广受欢迎，名气岂不全让我赚了？这人真傻。

七月的艳阳炙烤着大地，室内活动陡然增多。我要去滨江的 HZEcers 英语角分享出书经历，地点恰好就在网易旁边，于是邀请了他。没想到这哥们儿把女朋友带了来，两人在活动中一前一后，紧张地用生涩的英语做自我介绍：“Good afternoon everyone, my name is Liu Peiqing...” 他的生物专业小女友梳着齐刘海，戴着圆框眼镜，英语比他溜得多：“...recently I'm learning programming, because I want to have more common topics with my boyfriend...” 单纯的出发点，真是傻得可爱。也只有这样的傻姑娘，才会爱上这样的傻小子吧？

没过多久，我的论坛上出现了一个名为 MonkeyDev 的工具，是刘培庆写的，号称“原有 iOSOpenDev 的升级，非越狱插件开发集成神器”。我大致浏览了一下，看上去挺复杂，技术含量挺高，但竟然是开源的?！从 Git 提交记录来看，刘培庆一直在花时间维护它，免费供大家使用。这个年头，还有这样的雷锋，做这样的好事？这人真傻。

金秋十月，刘培庆又在微信上找我，说他要出书了，邀请我写个序。坦白地说，以刘培庆、

James 等 90 后为代表的新一代 iOS 逆向工程师鼓捣出来的新技术，我其实看不懂，也跟不上；同时，因为对库克的失望，我早已不再往苹果系技术上投入更多的精力。我已然从原来技术舞台的主角沦为了看客。“后浪”竟然找已经“死在沙滩上”的“前浪”写序？这人真傻。

但是，傻子刘培庆就在这样的评价中朝着自己的目标一步步前进，最终写出了《iOS 应用逆向与安全》，推动了行业的发展。

他让我想起了行业内的其他傻子。

四年前，另一个傻子，不知天高地厚地出版了《iOS 应用逆向工程》，填补了市场的空白，受到了读者的欢迎。这个傻子办了个名叫 iOSRE 的论坛，为所有 iOS 逆向工程爱好者提供免费、自由、平等、纯净的交流平台，却不做广告、不收赞助，自掏腰包维护论坛。这人真傻。

一个来自加拿大的傻子，为越狱 iOS 写了个名叫 Activator 的插件，全球总下载量近 2 亿次——据说这是乔布斯最喜爱的越狱插件。如果每次下载只收 1 毛钱，这个傻子也能成为千万富翁，可是他却把 Activator 免费提供给大家使用。这人真傻。

一个大学辍学的美国傻子，为越狱 iOS 提供了一套名为 Theos 的开发工具，它的简单易用吸引了大量人才进入这个领域，为越狱开发的黄金 5 年揭开了序幕。为了维护这套免费、开源的工具，他每天熬夜到凌晨 2 点，义务解决用户的问题，优化它的体验，却分文不取。这人真傻。

一个被亲生父母遗弃的傻子，都没正经上过大学，就自不量力地想要“Think different”。他创造的产品改变了世界，却积劳成疾，英年早逝，留给后人一句“Stay hungry, stay FOOLISH”。这人真傻。

“……

我不害怕全世界就剩下我一个傻瓜

我要坚持到底，用我的方式

别在意这世界的奇妙

……”

感谢刘培庆这样的傻子们。世界因为你们，变得可爱了一些。

沙梓社

2018 年 3 月 28 日夜，于杭州

前　言

2015 年，通过校招，我以 Windows 安全方向进入网易，组内安排投入 iOS 安全方向的研究。当时，我连苹果产品都没用过，于是攒了点钱，在淘宝上买了一台可越狱的 iPad 来研究。因为之前也没有接触过 Objective-C，无法深入阅读当时在网上找到的教程，所以只能跟着敲敲代码，看看效果。后来，通过研读念茜的文章，以及国外博客上的一些教程，把基本工具实践了一番，“狗神”沙梓社的书出版后，把他的书认真看了一遍，才算是踏进了门槛。

在那一段学习过程中，我对新的知识点都是囫囵吞枣，一直停留在工具的使用上，没有形成完整的知识体系和深层的认识，一旦出现问题就要花很长的时间去解决。这一点在后面张平引荐我去做网易云课堂的教学视频时感受尤为深刻——当你要规划整个课程时，你必须从全局出发考虑问题，仅仅根据自身的经验、知道工具的使用方法是不够的，只有理解和掌握原理，才能达到举一反三的效果。虽然从准备资料、制作 PPT 到最后录制视频的过程挺累的，但在这个过程中，我加深了对知识点的理解。这是我第一次录制视频课程，由于经验不足，导致了部分视频在终端的显示字体太小等问题，但总的来说，还是要感谢那些信任我、购买了我的视频课程的人。

后来，有几个朋友建议我出本书，把掌握的东西分享出来，也让新人少踩点坑。当时我是有点犹豫的。我不仅担心写书会占用很多时间，也担心自己的水平不够、写得不好。后来，想到视频里面的一些内容需要更新，很多知识点可以补充和完善，加上书籍的学习和沉淀效果也比视频好一些，我就开始做准备，规划每一章的完成时间，每天下班后或者周末在电脑前整理资料、写书，也挺充实的。现在，这本书终于和你见面了，希望书中的内容能够帮助你扩充自己的知识面，少走弯路，成为技术大牛。

读者对象

本书介绍了 iOS 开发、逆向和安全等方面的内容，面向以下读者：

- 高校计算机相关专业的学生

- iOS 开发工程师
- 逆向工程师
- 越狱开发工程师
- iOS 安全工程师
- 应用安全审计人员

近几年，iOS 开发人员数量激增。正向开发人员应该努力提升自己的竞争力，掌握一些底层技能，为自己开发的应用保驾护航。逆向新人也不要一味追求工具的使用和功能的实现，应该静下心来，基础知识掌握得扎实一些，后面的问题自然迎刃而解。

如何阅读本书

考虑到很多逆向分析人员缺乏正向开发和安全保护方面的知识，本书将分成以下 4 个部分进行讲解。

- 第 1 章 ~ 第 3 章是快速上手部分，内容包括一些基本概念的介绍，环境的准备，以及一些常用逆向分析工具的使用和原理。
- 第 4 章 ~ 第 6 章是正向知识储备和进阶部分，内容包括逆向过程中一些理论知识的深入讲解，例如类的结构、App 签名、Mach-O 文件格式、hook 原理等。
- 第 7 章是逆向实战部分，通过在越狱平台和非越狱平台上的逆向分析实例，带领读者学习逆向分析的思路和方法。
- 第 8 章是安全保护部分，内容包括应用安全及保护方面的知识，涉及网络传输加密、动态保护、代码混淆等。

尽管不同的人感兴趣的方面可能不一样，但我还是建议读者能够从头开始阅读本书，并把书中提到的每个知识点都实践一遍，以加深理解。

本书的源代码可以在 GitHub 上面找到：

https://github.com/AloneMonkey/iOSREBook

声明

本书的写作花费了大量的时间和心血，我只是想帮助大家在学习过程中少走弯路、拓宽知识面、增加技术积累，所以，请支持正版书籍，坚决抵制盗版！另外，本书内容仅供技术学习和研究之用，请勿将本书内容用于非法商业用途。

勘误

由于知识水平有限，写作过程也比较匆忙，书中难免出现错误及不足，欢迎各位读者指正。同时，我为本书开设了一个提交 issue 的项目：

https://github.com/AloneMonkey/iOSREBook-issues

致谢

感谢我的家人，在我成长的路上一直支持我、鼓励我。

感谢我异地三年的女友，很抱歉没有陪在你的身边。即便如此，你还是不离不弃，一直支持我的选择。

感谢念茜、狗神及在我的学习路上给予我帮助的人，感谢在网易期间的所有同事，是你们让我不断成长。

感谢电子工业出版社提供的平台，感谢编辑潘昕对本书内容的把控和指导。

感谢正在阅读本书的你，谢谢你的支持和信任。

刘培庆

2018 年 4 月，于杭州

目　录

第1章　概述

第2章　越狱设备

第3章 逆向工具详解

第4章 开发储备

第5章 分析与调试

第 6 章 逆向进阶

第 7 章 实战演练

第 8 章　安全保护

第1章　概述

1.1　逆向工程简介

iOS 应用逆向工程（后文简称“iOS 逆向”），是指从目标应用的界面及功能表现入手，使用不同的工具和理论知识去分析其实现原理，得出应用的代码结构、整体设计、功能实现、执行流程等，然后利用 iOS 的系统知识和语言特性，借鉴或修改原有实现流程的技术。如果你从未接触过逆向或者只接触过正向开发，你都将通过本书开启新的征程！

1.1.1　iOS 逆向学习基础

如果你是 iOS 正向开发者，你可以运用丰富的开发经验及对语言的了解，从界面功能知晓程序使用的系统组件或者第三方库，通过一些系统接口的调用及代理方法快速定位代码和发现线索。除此之外，你需要了解一些底层知识，例如文件结构、汇编知识、工具的使用及技巧。

如果你是其他平台的逆向者，并且熟悉该平台的文件格式（例如 PE、ELF），了解如何通过跳转表或 inline hook 的方式修改程序逻辑，那么这些知识可以帮助你快速学习 iOS 逆向。当然，你也需要了解 iOS 系统安全机制、语言特性及分析工具的原理。

如果你两者都不是，那也没有关系。只要保持积极学习的心态、锲而不舍的探索精神，通过逐步了解 iOS 系统机制、语言特性，不断实践，解决各种难题，并且在遇到问题时去探索问题的本质和原理，不断进行总结和积累，你也将成为一个逆向高手！

1.1.2　iOS 逆向的流程

在开始一系列的学习前，我们有必要了解 iOS 逆向工程的整个流程，以便对本书的内容及自己的学习进度有更好的把控。iOS 逆向工程的流程大致如下。

①解密、导出应用程序、class-dump 导出头文件，为后续工作做准备。

②从界面表现入手，获取当前界面布局及控制器。

③hook 发现的一些相关类，记录输出调用顺序及参数。

④找到关键函数，查看调用堆栈，hook 测试效果。

⑤静态分析加动态调试分析关键函数的实现逻辑。

⑥模拟或篡改函数调用逻辑。

⑦制作插件，或者移植到非越狱机器。

整体流程就是这样，但过程可能不会太顺利，有时需要反复探索才能定位目标函数。这个过程很枯燥，也很有趣。在多次分析和实践之后，你会找到属于自己的分析方法和技巧。

1.1.3 iOS 逆向使用的工具

对任何平台进行逆向分析时都会借助很多工具，iOS 逆向也不例外。以逆向流程为例，会使用解密工具、class-dump、Cycript、Reveal、Charles、Hopper、IDA、LLDB、Xcode、Theos 等，这些工具会在后面的章节中详细讲解。当然，仅学会工具的使用是远远不够的。在掌握工具使用方法的前提下，还需要了解每一个工具的实现原理是什么，有哪些可以借鉴的地方，如果是自己去实现应该怎么做等。本书在讲解工具的使用时，都会介绍工具的原理，从而帮助你查找出错的原因和改进现有的工具。

1.1.4 iOS 逆向的应用场景

很多人起初只是觉得 iOS 逆向工程很神秘，能做很多有意思的事情，才会去学习和探索。那么，学完 iOS 逆向之后到底可以做什么？能获得哪些帮助？笔者觉得有以下几点。

- 促进正向开发，深入理解系统原理。
- 借鉴别人的设计和实现，实现自己的功能。
- 分析恶意软件，应用安全审计。
- 从逆向的角度实现安全保护。

学完逆向工程之后，你将可以从不同的角度去看待和思考问题。

1.2 应用面临的安全风险

随着 iOS 逆向的工具越来越成熟，入门门槛越来越低，加上大部分应用都没有采取保护措施，现在市面上的很多应用都面临被破解、被篡改及协议被模拟等风险，也出现了不少多开、作弊、刷榜等恶意行为，应用面临的安全问题已经非常严峻。下面从 3 个方面分别举例说明。

1.2.1　静态修改文件

虽然在 iOS 平台上直接静态修改汇编的情况较少（当然也是可以的），但还是可以通过 insert_dylib 或者 optool 等工具对可执行文件进行注入，使其加载指定的动态库文件，达到篡改的效果（非越狱插件）。另外，可以通过修改本地读取的特定文件去修改程序的功能。

1.2.2　动态篡改逻辑

在越狱设备上，可以通过 Cydia Substrate 提供的注入和 hook 模块注入指定的程序，篡改程序的功能。在非越狱设备上，也可以静态注入动态库，使用 fishhook 或者 Method Swizzle 达到修改或增加功能的效果。例如，修改 Pokeman GO 的定位的信息及移动速度。再如，在某应用收到红包时，自动调用领取红包的功能函数，实现“秒抢”的效果。此外，可以解除会员限制、破解本地验证等。

1.2.3　协议分析

除了修改应用本身，还可以逆向分析应用程序的请求协议，并通过其他程序来模拟协议，实现脱机。这种机制经常被利用进行恶意注册和刷单等。

1.3　应用保护手段

有攻便有防。为了保护应用不被篡改、分析、重签名等，需要采取一定的保护措施。虽然安全保护只是增加了逆向分析的难度，但这种方法往往是有效且必要的。

1.3.1　数据加密

应用中一般都会存储很多敏感数据，例如加密和解密的 key、用户的隐私信息、通信传输的信息。如果开发者的安全意识比较弱，可能会把一些敏感信息或者敏感字符串以明文的形式写在代码里面。此外，传输的一些关键数据也不会加密，而这往往会给分析者提供很多关键信息，增加应用被破解的风险。针对数据加密，可以从以下 3 个方面入手。

- 静态字符串加密
- 本地存储加密
- 网络传输加密

通过静态字符串加密隐藏关键信息，可以增加静态分析的难度。对本地和网络数据进行加

密，也可以避免重要信息泄露和中间人攻击。

1.3.2 程序混淆

程序中的类名和方法名定义往往是有具体含义的，攻击者根据方法名就能分析出该方法的用途。为了避免这种情况，可以把类名和方法名替换成无意义的随机字符串，让分析者无法通过名字获取有意义的信息。

除此之外，使用静态分析工具进行反编译是很容易看出程序的实现逻辑和调用关系的，所以，对关键代码的保护和混淆不可或缺。可以通过 LLVM 在编译过程中对中间代码进行控制流程的混淆、代码扁平化、插入干扰代码等操作，以干扰分析者的静态分析，增加分析的难度和成本。

1.3.3 安全监测

分析者在分析应用时，可以通过调试、注入动态库、重签名等手段修改原有应用。为保证应用运行时的安全性，可以在应用中加入反调试代码，检测调试器，或者加入动态库注入检测、函数 hook 检测、签名检测、完整性等，使得当检测到应用被侵入时，程序退出或者功能运行出现异常。

通过以上对 iOS 逆向工程的大致介绍，相信你对逆向工程的内容及学习流程都有了一定的了解。在学习过程中，要着重于理论基础和动手实践，慢慢地领会逆向的魅力。无论你是安全人员、开发者还是逆向新手，都能从中感受到乐趣。同时，正向开发者和安全人员对本章提到的风险也要认真对待，因为在学习逆向分析的方法之后，就能针对不同的分析方法提供不同的安全保护方法，从而进一步保护应用的安全了。

1.4 本书工具

为了让你更加快速地上手，也为了避免后续对一些 Mac 常用工具的重复介绍，本节将对本书中使用的一些 Mac 工具进行总体介绍。你不用马上把下面所有的工具都安装好，可以在后续使用时再安装。

1.4.1 效率工具

在工作中，一个好的开发环境可以帮助我们节约很多时间。正所谓“工欲善其事，必先利

其器”，先把利器准备好，才能高效地进行分析。下面介绍几款能够提高 Mac 开发分析效率的工具。

- iTerm2：代替默认的 Terminal，提供了很多高级设置，例如自动补全、高亮等（https://iterm2.com/）。
- oh-my-zsh：可以自定义主题、Git 显示、Tab 补全等（https://github.com/robbyrussell/oh-my-zsh）。
- Go2Shell：从 Finder 打开终端并自动切换到当前目录（http://zipzapmac.com/go2shell）。
- autojump：从终端快速进行目录跳转和切换（https://github.com/wting/autojump）。
- Alfred：快速打开软件，自定义脚本执行（https://www.alfredapp.com/）。

1.4.2　实用工具

效率提升后，需要一些实用的工具来帮助完成某些复杂的操作。下面来看几款实用的工具。

- Homebrew：Mac OS 的包管理工具，可以快速安装各种工具（https://brew.sh/）。
- Cakebrew：Homebrew 的界面管理工具（https://www.cakebrew.com/）。
- libimobiledevice：提供了很多能与 iOS 交互的工具，例如端口映射查看日志（https://github.com/libimobiledevice）。
- tree：用于查看当前目录结构树的命令行工具，通过 brew install tree 安装。
- 010 Editor：二进制编辑分析工具（https://www.sweetscape.com/010editor/）。

1.4.3　逆向工具

关于在逆向分析中使用的工具，在后面的章节里会进行详细讲解。本书没有深入讲解的工具在这里介绍如下。

- jtool：查看文件结构、代码签名（http://www.newosxbook.com/tools/jtool.html）。
- capstone：多平台、多架构支持的反汇编框架（http://www.capstone-engine.org/）。
- keystone：将汇编指令转换为 Hex 机器码（https://github.com/keystone-engine/keystone）。
- radare2：一款开放源代码的逆向工程平台（https://github.com/radare/radare2）。
- mobiledevice：安装 app 或 ipa 包（https://github.com/imkira/mobiledevice）。