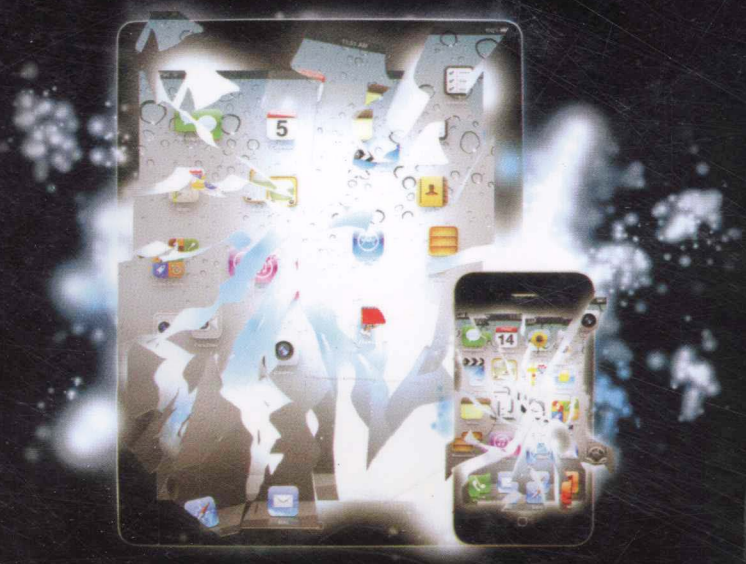
TURING 图灵程序设计丛书 网络安全系列

iOS Hacker's Handbook

黑客攻防技术宝典

iOS实战篇

[美] Charlie Miller Dionysus Blazakis Dino Dai Zovi
Stefan Esser Vincenzo Iozzo Ralf-Philipp Weinmann 著
傅尔也 译



美国国家安全局全球网络漏洞攻击分析师、连续4年Pwn2Own黑客竞赛大奖得主Charlie Miller主笔

作者阵容超级豪华，6位均为信息安全领域大名鼎鼎的顶级专家，各有所长，且多有专著出版

国内唯一专注iOS平台漏洞、破解及安全攻防的中文专著

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 网络安全系列

iOS Hacker's Handbook

黑客攻防技术宝典

iOS实战篇

[美] Charlie Miller Dionysus Blazakis Dino Dai Zovi
Stefan Esser Vincenzo Iozzo Ralf-Philipp Weinmann 著
傅尔也 译

人民邮电出版社
北京

图书在版编目（CIP）数据

黑客攻防技术宝典. iOS实战篇 /（美）米勒(Miller,C.）等著 ；傅尔也译. -- 北京 ：人民邮电出版社, 2013.9
（图灵程序设计丛书）
书名原文: iOS hacker's handbook
ISBN 978-7-115-32848-9

Ⅰ. ①黑… Ⅱ. ①米… ②傅… Ⅲ. ①计算机网络－安全技术 Ⅳ. ①TP393.08

中国版本图书馆CIP数据核字(2013)第187413号

内 容 提 要

《黑客攻防技术宝典：iOS 实战篇》全面介绍 iOS 的安全性及工作原理，揭示了可能威胁 iOS 移动设备的所有安全风险和漏洞攻击程序，致力于打造一个更安全的平台。本书内容包括：iOS 设备和 iOS 安全架构、iOS 在企业中的应用（企业管理和服务提供）、加密敏感数据的处理、代码签名、沙盒的相关机制与处理、用模糊测试从默认 iOS 应用中查找漏洞、编写漏洞攻击程序、面向返回的程序设计（ROP）、iOS 内核调试与漏洞审查、越狱工作原理与工具、基带处理器。

本书适合所有希望了解 iOS 设备工作原理的人学习参考，包括致力于以安全方式存储数据的应用开发人员、保障 iOS 设备安全的企业管理人员、从 iOS 中寻找瑕疵的安全研究人员，以及希望融入越狱社区者。

◆ 著　[美] Charlie Miller　Dionysus Blazakis　Dino Dai Zovi　Stefan Esser　Vincenzo Iozzo　Ralf-Philipp Weinmann
译　傅尔也
责任编辑　毛倩倩
责任印制　焦志炜

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街14号
邮编　100061　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：20
字数：478千字　2013年9月第1版
印数：1-4 000册　2013年9月北京第1次印刷
著作权合同登记号　图字：01-2012-5235号

定价：69.00元

读者服务热线：(010)51095186转604　印装质量热线：(010)67129223
反盗版热线：(010)67171154
广告经营许可证：京崇工商广字第0021号

版权声明

作者介绍

Charlie Miller Accuvant Labs 首席研究顾问，曾在美国国家安全局担任全球网络漏洞攻击分析师 5 年，连续 4 年赢得 CanSecWest Pwn2Own 黑客大赛。他发现了 iPhone 与 G1 安卓手机第一个公开的远程漏洞，通过短信对 iPhone 进行漏洞攻击并发现了可以让恶意软件进入 iOS 的代码签名机制缺陷。作为圣母大学博士的他还与人合著了 *The Mac Hacker's Handbook* 和 *Fuzzing for Software Security Testing and Quality Assurance* 两本信息安全类图书。

Dionysus Blazakis 程序员和安全研究人员，擅长漏洞攻击缓解技术，经常在安全会议上发表有关漏洞攻击缓解技术、绕过缓解技术和寻找漏洞的新方法等主题演讲，因利用即时编译器绕过数据执行保护的技术赢得了 2010 年 Pwnie Award 最具创新研究奖。另外，他与 Charlie Miller 为参加 2011 年 Pwn2Own 大赛开发的 iOS 漏洞攻击程序赢得了 iPhone 漏洞攻击比赛的大奖。

Dino Dai Zovi Trail of Bits 联合创始人和首席技术官，有十余年信息安全领域从业经验，做过红队（red teaming，又称“伦理黑客”）、渗透测试、软件安全、信息安全管理和网络安全研究与开发等多种工作。Dino 是信息安全会议的常客，在 DEFCON、BlackHat 和 CanSecWest 等世界知名的信息安全会议上发表过对内存损坏利用技术、802.11 无线客户端攻击和英特尔 VT-x 虚拟化 rootkit 程序等课题的独立研究成果。他还是 *The Mac Hacker's Handbook* 和 *The Art of Software Security Testing* 的合著者。

Vincenzo Iozzo Tiqad srl 安全研究人员，BlackHat 和 Shakacon 安全会议评审委员会成员，常在 BlackHat 和 CanSecWest 等信息安全会议上发表演讲。他与人合作为 BlackBerryOS 和 iPhoneOS 编写了漏洞攻击程序，因 2010 年和 2011 年连续两届获得 Pwn2Own 比赛大奖在信息安全领域名声大振。

Stefan Esser 因在 PHP 安全方面的造诣为人熟知，2002 年成为 PHP 核心开发者以来主要关注 PHP 和 PHP 应用程序漏洞的研究，早期发表过很多关于 CVS、Samba、OpenBSD 或 Internet Explorer 等软件中漏洞的报告。2003 年他利用了 XBOX 字体加载器中存在的缓冲区溢出漏洞，成为从原厂 XBOX 的硬盘上直接引导 Linux 成功的第一人；2004 年成立 Hardened-PHP 项目，旨在开发更安全的 PHP，也就是 Hardened-PHP（2006 年融入 Suhosin PHP 安全系统）；2007 年与人

合办德国 Web 应用开发公司 SektionEins GmbH 并负责研发工作；2010 年起积极研究 iOS 安全问题，并在 2011 年提供了一个用于越狱的漏洞攻击程序（曾在苹果多次更新后幸存下来）。

Ralf-Philipp Weinmann　德国达姆施塔特工业大学密码学博士、卢森堡大学博士后研究员。他在信息安全方面的研究方向众多，涉及密码学、移动设备安全等很多主题。让他声名远播的事迹包括参与让 WEP 破解剧烈提速的项目、分析苹果的 FileVault 加密、擅长逆向工程技术、攻破 DECT 中的专属加密算法，以及成功通过智能手机的 Web 浏览器（Pwn2Own）和 GSM 协议栈进行渗透攻击。

前　言

iPhone 已经问世 5 年有余，人们大概都已经忘了当时的 iPhone 多么具有开创意义。那时候还没有现在这样的智能手机，很多手机也就是用来打打电话。有些手机中安装了 Web 浏览器，但并非全功能的，只能呈现最基本的网页，而且手机屏幕的分辨率非常低。好在，iPhone 改变了这一切。

iPhone 的显示屏几乎占据整个前面板，有着基于 WebKit 的 Web 浏览器，而且其操作系统可以由用户自行升级，不需要等着运营商来做这项工作。再加上存储照片、播放音乐和发送短信等功能，这才是人们真正想要拥有的手机（参见图 1）。但是，iPhone 并不完美。第一代 iPhone 数据传输速度非常慢，不支持第三方应用，而且安全性特别差，不过它却引领了智能手机和平板电脑的革命。

图 1　众多消费者排队等待购买第一代 iPhone

图片来源：Mark Kriegsman（http://www.flickr.com/photos/kriegsman/663122857/）。

随着第一代 iPhone 于 2007 年问世，一系列其他的苹果产品也随之而来，而它们都运行着 iOS。当然，在第一代 iPhone 等设备问世时这个操作系统还不叫 iOS。第一代 iPhone 使用的操作系统被苹果公司称为 OS X，就像其桌面版的“兄长”那样。而在 2008 年第二代 iPhone 出现时，这个操作系统被称为 iPhone OS。那时候它还不能拥有 iOS 这个称呼，因为思科公司为路由器设计的操作系统先占用了 IOS 这个名称。经过一番交易，苹果公司从 2010 年起正式将其移动操作系统命名为 iOS。

紧随 iPhone 之后的 iOS 设备是 iPod Touch。这种设备基本上就是个不能打电话不能发短信的 iPhone。其他 iOS 设备包括第二代 Apple TV 和 iPad。这些设备每推出新的一代，都是更快、更时髦、更多功能的产品（如图 2 所示）。

图 2　iPhone 4（左）与 iPhone 1（右）的对比

全书概览

不过，人们通常只注意这些设备光鲜的外表，很少会去了解它们的内部工作原理。数百万人每天随身携带存放着他们个人信息的这些小设备，但它们到底安全吗？在各种安全大会的演讲中，在越狱社区里，甚至在研究人员的个人日志中，我们都可以发现关于 iOS 运行安全的信息。本书就是要把这些有关 iOS 内部原理的知识汇总起来。只有让人们都能接触到这些信息，才能让个人和企业有效评估使用这些设备的风险，并了解如何最大限度地降低这种风险。本书甚至可以提供一些让设备本身更安全与让用户使用起来也更安全的思路。

本书内容

本书是按 iOS 安全功能主题划分章节的，读者可以用不同的方式来阅读本书。不熟悉这些主题或是不想错过任何内容的读者可以从头至尾阅读整本书。本书从相对基础的章节开始，由浅入深地慢慢过渡到后面较为复杂和深奥的章节。而那些已经对 iOS 的内部细节有所了解的读者可以

跳过开头部分，直接阅读自己感兴趣的那些章节。每一章的内容基本上都是相对独立的。在提到其他章的主题时，我们都会指明出处。下面来看一下本书中各章的主要内容。

- **第 1 章**概述 iOS 设备和 iOS 安全架构。我们在此介绍本书其余部分所要讨论的大部分主题，最后讨论针对各版 iOS 发动的一些攻击，包括最早期的一些攻击和针对 iOS 5 安全架构的一些攻击。
- **第 2 章**讨论 iOS 在企业中的使用，涉及诸如企业管理和服务提供之类的主题。此外，这一章还讲述如何为企业设备开发应用，包括开发者证书和配置概要文件的工作原理。
- **第 3 章**包含与 iOS 处理加密敏感数据相关的信息。这一章概述如何为每台 iOS 设备得出加密密钥以及如何使用这些加密密钥、各种等级的加密以及每种等级下都有哪些文件，讨论开发人员如何利用 Data Protection API 保护应用中的敏感数据。最后，我们还将展示如何通过蛮力攻击破解密码，以及 4 位数字密码的脆弱性。
- **第 4 章**针对 iOS 深入介绍一种主要的安全机制——代码签名。我们将为读者呈现相关的源代码和逆向工程二进制文件，它们用于确保只有由受信任机构签名的代码才能在设备上运行。这一章还将重点介绍 iOS 代码签名机制中的新内容，它们为实现即时编译而允许未签名的代码以一种严格受控的方式运行。最后，我们介绍 iOS 5 的早期版本中出现的代码签名机制漏洞。
- **第 5 章**介绍 iOS 中涉及沙盒的机制。我们将展示 iOS 内核如何支持把钩子程序放置在关键区域，讨论沙盒具体用到的钩子，然后举例说明应用如何完成自己的沙盒处理，并讲述重要的 iOS 功能是如何执行沙盒处理的。最后，这一章将讨论沙盒描述文件、这些文件如何描述沙盒所许可的功能，以及如何从 iOS 二进制文件中提取这些文件以用于研究。
- **第 6 章**展示如何利用模糊测试技术从默认的 iOS 应用中找到漏洞。我们首先综合探讨模糊测试，接着展示如何对 iOS 中最大的受攻击面 MobileSafari 进行模糊测试。这一章重点介绍进行 iOS 模糊测试的几种不同方式，包括在 Mac OS X、iOS 模拟器以及 iOS 设备上进行模糊测试。最后，我们还将展示如何对台式机上没有的 SMS 解析器进行模糊测试。
- **第 7 章**讲述如何利用第 6 章介绍的技术找到漏洞，并将其转换为有效的漏洞攻击程序。我们将详细分析 iOS 的堆管理系统，并说明如何利用“堆风水”技术操控堆内存。然后，这一章讨论漏洞攻击程序开发中的一个主要障碍——地址空间布局随机化（ASLR）。
- **第 8 章**进一步向大家展示在控制进程后可以做些什么。在简要介绍 iOS 设备中使用的 ARM 架构后，我们就转而介绍面向返回的程序设计（ROP）。这里将向大家介绍如何手工创建和自动生成 ROP 有效载荷，还将给出一些 ROP 有效载荷的例子。
- **第 9 章**从用户空间转入内核。在介绍一些内核基础知识后，我们接着描述如何调试 iOS 内核从而监控其动态。这一章还将展示如何对内核进行漏洞审查以及如何利用找到的各种漏洞。
- **第 10 章**介绍越狱。首先，这一章讲述有关越狱工作原理的基础知识，接着详细描述不同类型的越狱工具，然后概述越狱工具所需的不同组成部分，包括对文件系统的修改、已安装的守护进程、激活，最后还将通览越狱利用的所有内核补丁。

- 第 11 章介绍很多 iOS 设备中都有的另一个处理器——基带处理器。我们将展示如何设置与基带进行交互的工具，并介绍从过去到现在 iOS 设备的基带中都使用了哪些实时操作系统，然后说明如何对基带操作系统进行审计，还给出了一些漏洞示例。最后，这一章还将描述一些可以在基带操作系统上运行的有效载荷。

读者对象

本书是为所有希望了解 iOS 设备工作原理的人所写的。他们可以是希望融入越狱社区的人，也可以是试图了解如何以安全方式存储数据的应用开发人员，还可以是想要了解如何保障 iOS 设备安全的企业管理人员，或者尝试从 iOS 中寻找瑕疵的安全研究人员。

这些目标读者几乎都应该阅读和理解本书前面的章节。虽然后面的章节也都试着从基础知识开始介绍，但是理解这些内容至少能熟悉一些基本套路，比方说如何使用调试器和如何阅读代码清单等。

所需工具

如果大家只想对 iOS 的工作原理有个初步的了解，本书完全可以满足需要。不过，为了掌握本书的绝大部分内容，我们希望大家参照书中示例在自己的 iOS 设备上进行操作。这样的话，大家就至少需要一部 iOS 设备。为了真正掌握这些例子，大家需要为 iOS 设备越狱。此外，虽然有可能为其他平台凑齐一套能起作用的工具，但是为了使用 Xcode 编译示例程序，大家最好有一台运行 Mac OS X 的计算机。

配套网站

本书配套网站 www.wiley.com/go/ioshackershandbook 中有本书的所有代码[①]，因此大家不需要自己一行一行敲代码。此外，对于书中提到的 iOS 特有的工具，只要有可能我们就都会收录在该网站上。本书勘误也可在本网站上查询，如果大家发现本书的错漏之处，还望不吝赐教。

祝贺大家

我们喜爱自己的 iOS 设备，我们都是果粉。不过，要是攻击者不能从中窃取个人信息的话，我们会更喜欢这些设备。尽管阅读本书这样的书籍没法让大家阻止所有针对 iOS 的攻击，但只有越来越多的人了解 iOS 的安全性及其工作原理，iOS 才可能成为一个更安全的平台。请大家准备好，我们马上就要探索 iOS 安全了，而且要努力让它变得更安全。毕竟，有所了解就等于成功了一半。

① 本书源代码也可在图灵社区本书网页（http://www.ituring.com.cn/book/1068）免费注册下载。——编者注

致　谢

我想感谢我的妻子 Andrea，感谢她的绵绵爱意与不断支持，还要谢谢我亲爱的儿子 Theo 和 Levi，他们将是 iOS 黑客和越狱界的新生代力量。

——Charlie

首先，我要感谢我的家人 Alayna、Simon 和 Oliver，感谢这几个月来我每晚下班回家后加班加点工作时他们所给予的耐心与关爱。我还想感谢越狱社区提供的种种帮助。他们除了开发专业的越狱工具，还提供了很多能让安全研究人员的工作变得更加简单的文档（比如 iPhone wiki），以及用于提取和修改 iOS 固件的工具。

——Dionysus

我要感谢我的父母、妹妹以及密友对我的不断支持，特别是在我参与编写此书的这段时间；没有他们的话，我想我早疯了。我还要感谢 iOS 越狱工具开发社区，感谢社区成员进行了大量的技术研究并免费发布开发出的工具，他们还常常提供全部的源代码。最后，我还要感谢 Pablo 和 Paco 在我上次写书时提供的帮助。

——Dino

我想感谢我的双亲、哥哥和各位密友，感谢他们总是支持我，哪怕我偶尔冒出疯狂的想法。另外，我还要特别感谢我多年来的灵魂伴侣 Nami。

——Stefan

我想感谢在我个人生活和专业领域中，每一个帮助我沿着这条坎坷之途一路走来的人。我想感谢的人实在太多，真的没办法在这里一一列出。我特别感谢在编写本书时助我一臂之力的 Naike 和 Max。

——Vincenzo

我想感谢我的妻女，因为她们长久以来不得不忍受我在写作时对她们视若无睹。我要感谢Thomas Dullien、Joshua Lackey和Harald Welte，在2010年我研究基带的几个月中，我们进行了很多富有启发性的探讨。非常感谢Jacob Appelbaum，他让我接触到了发起我要研究的主题的那些工程师。我还要对那些不愿留名的幕后英雄表示感谢，他们知道我说的是谁，感谢他们所做的一切！最后我要感谢iPhone Dev Team所做的工作，要是没有他们的成果，很多事情就要难办很多。在此，我特别感谢MuscleNerd（肌肉男）和planetbeing在我被iPhone4难住时提供的帮助，还要感谢roxfan为我提供了他的分散加载脚本。

——Ralf

目　录

第 1 章　iOS 安全基础知识 ……1
1.1　iOS 硬件/设备的类型 ……1
1.2　苹果公司如何保护 App Store ……2
1.3　理解安全威胁 ……3
1.4　理解 iOS 的安全架构 ……4
1.4.1　更小的受攻击面 ……4
1.4.2　精简过的 iOS ……5
1.4.3　权限分离 ……5
1.4.4　代码签名 ……5
1.4.5　数据执行保护 ……6
1.4.6　地址空间布局随机化 ……6
1.4.7　沙盒 ……6
1.5　iOS 攻击简史 ……7
1.5.1　Libtiff ……7
1.5.2　短信攻击 ……8
1.5.3　Ikee 蠕虫 ……8
1.5.4　Storm8 ……9
1.5.5　SpyPhone ……10
1.5.6　Pwn2Own 2010 ……10
1.5.7　Jailbreakme.com 2 ("Star") ……10
1.5.8　Jailbreakme.com 3 ("Saffron") ……11
1.6　小结 ……11

第 2 章　企业中的 iOS ……12
2.1　iOS 配置管理 ……12
2.1.1　移动配置描述文件 ……13
2.1.2　iPhone 配置实用工具 ……14
2.2　移动设备管理 ……21
2.2.1　MDM 网络通信 ……21
2.2.2　Lion Server 描述文件管理器 ……22
2.3　小结 ……36

第 3 章　加密 ……37
3.1　数据保护 ……37
3.2　对数据保护的攻击 ……40
3.2.1　对用户密码的攻击 ……40
3.2.2　iPhone Data Protection Tools ……43
3.3　小结 ……54

第 4 章　代码签名和内存保护 ……55
4.1　强制访问控制 ……56
4.1.1　AMFI 钩子 ……56
4.1.2　AMFI 和 execv ……57
4.2　授权的工作原理 ……59
4.2.1　理解授权描述文件 ……59
4.2.2　如何验证授权文件的有效性 ……62
4.3　理解应用签名 ……62
4.4　深入了解特权 ……64
4.5　代码签名的实施方法 ……65
4.5.1　收集和验证签名信息 ……65
4.5.2　如何在进程上实施签名 ……68
4.5.3　iOS 如何确保已签名页不发生改变 ……72
4.6　探索动态代码签名 ……73
4.6.1　MobileSafari 的特殊性 ……73
4.6.2　内核如何处理即时编译 ……75
4.6.3　MobileSafari 内部的攻击 ……77
4.7　破坏代码签名机制 ……78
4.7.1　修改 iOS shellcode ……79
4.7.2　在 iOS 上使用 Meterpreter ……83

4.7.3 取得 App Store 的批准 ……85
4.8 小结 ……86
第 5 章 沙盒 ……87
5.1 理解沙盒 ……87
5.2 在应用开发中使用沙盒 ……89
5.3 理解沙盒的实现 ……95
5.3.1 理解用户空间库的实现 ……95
5.3.2 深入内核 ……98
5.3.3 沙盒机制对 App Store 应用和平台应用的影响 ……109
5.4 小结 ……113
第 6 章 对 iOS 应用进行模糊测试 ……114
6.1 模糊测试的原理 ……114
6.2 如何进行模糊测试 ……115
6.2.1 基于变异的模糊测试 ……116
6.2.2 基于生成的模糊测试 ……116
6.2.3 提交和监测测试用例 ……117
6.3 对 Safari 进行模糊测试 ……118
6.3.1 选择接口 ……118
6.3.2 生成测试用例 ……118
6.3.3 测试和监测应用 ……119
6.4 PDF 模糊测试中的冒险 ……122
6.5 对快速查看（Quick Look）的模糊测试 ……126
6.6 用模拟器进行模糊测试 ……127
6.7 对 MobileSafari 进行模糊测试 ……130
6.7.1 选择进行模糊测试的接口 ……130
6.7.2 生成测试用例 ……130
6.7.3 MobileSafari 的模糊测试与监测 ……131
6.8 PPT 模糊测试 ……133
6.9 对 SMS 的模糊测试 ……134
6.9.1 SMS 基础知识 ……135
6.9.2 聚焦协议数据单元模式 ……136
6.9.3 PDUspy 的使用 ……138
6.9.4 用户数据头信息的使用 ……139
6.9.5 拼接消息的处理 ……139
6.9.6 其他类型 UDH 数据的使用 ……139
6.9.7 用 Sulley 进行基于生成的模糊测试 ……141
6.9.8 SMS iOS 注入 ……145
6.9.9 SMS 的监测 ……146
6.9.10 SMS bug ……151
6.10 小结 ……153
第 7 章 漏洞攻击 ……154
7.1 针对 bug 类的漏洞攻击 ……154
7.2 理解 iOS 系统自带的分配程序 ……156
7.2.1 区域 ……156
7.2.2 内存分配 ……157
7.2.3 内存释放 ……157
7.3 驯服 iOS 的分配程序 ……158
7.3.1 所需工具 ……158
7.3.2 与分配/释放有关的基础知识 ……159
7.4 理解 TCMalloc ……167
7.4.1 大对象的分配和释放 ……167
7.4.2 小对象的分配 ……168
7.4.3 小对象的释放 ……168
7.5 驯服 TCMalloc ……168
7.5.1 获得可预知的堆布局 ……168
7.5.2 用于调试堆操作代码的工具 ……170
7.5.3 堆风水：以 TCMalloc 对算术漏洞进行攻击 ……172
7.5.4 以 TCMalloc 就对象生存期问题进行漏洞攻击 ……175
7.6 对 ASLR 的挑战 ……176
7.7 案例研究：Pwn2Own 2010 ……177
7.8 测试基础设施 ……181
7.9 小结 ……181
第 8 章 面向返回的程序设计 ……182
8.1 ARM 基础知识 ……182
8.1.1 iOS 的调用约定 ……183
8.1.2 系统调用的调用约定 ……183
8.2 ROP 简介 ……185
8.2.1 ROP 与堆 bug ……186
8.2.2 手工构造 ROP 有效载荷 ……187
8.2.3 ROP 有效载荷构造过程的自动化 ……191

8.3 在 iOS 中使用 ROP……193
8.4 iOS 中 ROP shellcode 的示例……195
8.4.1 用于盗取文件内容的有效载荷……196
8.4.2 利用 ROP 结合两种漏洞攻击程序（JailBreakMe v3）……202
8.5 小结……206

第 9 章 内核的调试与漏洞攻击……207
9.1 内核的结构……207
9.2 内核的调试……208
9.3 内核扩展与 IOKit 驱动程序……213
9.3.1 对 IOKit 驱动程序对象树的逆向处理……213
9.3.2 在内核扩展中寻找漏洞……216
9.3.3 在 IOKit 驱动程序中寻找漏洞……219
9.4 内核漏洞攻击……222
9.4.1 任意内存的重写……223
9.4.2 未初始化的内核变量……227
9.4.3 内核栈缓冲区溢出……231
9.4.4 内核堆缓冲区溢出……236
9.5 小结……245

第 10 章 越狱……246
10.1 为何越狱……246
10.2 越狱的类型……247
10.2.1 越狱的持久性……247
10.2.2 漏洞攻击程序的类型……248
10.3 理解越狱过程……249
10.3.1 对 bootrom 进行漏洞攻击……250
10.3.2 引导 ramdisk……250
10.3.3 为文件系统越狱……250
10.3.4 安装完美越狱漏洞攻击程序……251
10.3.5 安装 AFC2 服务……251
10.3.6 安装基本实用工具……252
10.3.7 应用转存……253
10.3.8 应用包安装……254
10.3.9 安装后的过程……255
10.4 执行内核有效载荷和补丁……255
10.4.1 内核状态修复……255
10.4.2 权限提升……256
10.4.3 为内核打补丁……257
10.4.4 安全返回……267
10.5 小结……268

第 11 章 基带攻击……269
11.1 GSM 基础知识……270
11.2 建立 OpenBTS……272
11.2.1 硬件要求……272
11.2.2 OpenBTS 的安装和配置……273
11.3 协议栈之下的 RTOS……276
11.3.1 Nucleus PLUS……276
11.3.2 ThreadX……277
11.3.3 REX/OKL4/Iguana……277
11.3.4 堆的实现……278
11.4 漏洞分析……281
11.4.1 获得并提取基带固件……281
11.4.2 将固件镜像载入 IDA Pro……283
11.4.3 应用/基带处理器接口……283
11.4.4 栈跟踪与基带核心转储……283
11.4.5 受攻击面……284
11.4.6 二进制代码的静态分析……285
11.4.7 由规范引路的模糊测试……285
11.5 对基带的漏洞攻击……286
11.5.1 本地栈缓冲区溢出：AT+XAPP……286
11.5.2 ultrasn0w 解锁工具……287
11.5.3 空中接口可利用的溢出……293
11.6 小结……299

附录 参考资料……300

第1章 iOS安全基础知识

如果你也像我们一样，那么只要拿到新设备就会想要了解它的安全性。这里的“设备”当然也包括iPhone。它不再只是带有小型Web浏览器的手机，与老式手机相比，它更像是计算机。当然，这些（以及将来的）设备可能存在与台式机中相似的安全问题。为了避免这些设备受到危害，苹果公司为它们内置了怎样的预防措施和安全机制呢？我们眼前是一个开启计算领域全新分支的机会。安全性对这些新兴智能设备来说有多重要呢？

本章会就iOS设备回答这些问题。首先，我们要看看各种iOS设备上使用的硬件，然后介绍iOS 5的安全架构。重点讲一讲现有设备为了防范恶意软件攻击和攻击者利用漏洞所内嵌的多层防御手段。接着，介绍一些已经发生的针对iOS设备的攻击，从而说明这些防御手段在现实世界中是如何起效（或失效）的。还将按照时间先后顺序，介绍从最早的iPhone到iOS 5设备受到的各种攻击。阅读过程中，大家会看到iOS设备的安全性有了多大的提高。最初版本的iOS几乎没有安全性可言，但iOS 5相对而言则既强大又可靠。

1.1 iOS 硬件/设备的类型

几年来，iOS一直在发展，各种苹果设备中的硬件也不断推陈出新。随着智能手机和平板电脑的普及，人们都希望拥有一台强大的计算设备。从某种意义上讲，他们期望自己口袋里装着的是一台电脑。

iPad的问世就是在这一方向上迈出的第一步。第一代iPad使用了ARM Cortex-A8架构的CPU，它的速度大约是第一代iPhone所使用CPU速度的两倍。

iPad 2和iPhone 4S则是另一个巨大跨越。它们都使用了ARM Cortex-A9架构的双核处理器，就CPU运算的速度而言，要比A8架构的处理器快20%。更惊人的是，A9的GPU要比A8的快9倍。

从安全的角度看，硬件上差异最大的是iPhone 3GS和iPad 2。iPhone 3GS是第一种支持Thumb2指令集的设备。这种新型指令集改变了创建ROP有效载荷的方式。之前设备中出现的代码序列在iPhone 3GS中突然发生了改变。

另一方面，iPad 2使用了双核处理器，它让iOS的分配程序可以全力运行。这样就对漏洞攻击的构造带来了巨大影响，因为漏洞攻击在多处理器环境下的可靠性要弱很多。