

<<iOS应用安全攻防>>

图书基本信息

书名：<<iOS应用安全攻防>>

13位ISBN编号：9787564134464

10位ISBN编号：7564134461

出版时间：2012-6

出版时间：东南大学出版社

作者：扎德爾斯基

页数：336

字数：436000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<iOS应用安全攻防>>

内容概要

如果你是一位具有坚实Objective-C基础的应用开发者，这本《iOS应用安全攻防(影印版)》绝对急你所需——你所在公司的iOS应用被攻击的可能性很大。

这是因为恶意攻击者现在使用一系列工具采用大多数程序员想象不到的方式进行反向工程、跟踪和操纵应用。

这本书讲解了几种iOS的攻击手段，以及黑客们常用的工具和技术。

你会从中学到保护你的应用的最佳方式，并且意识到像你的对手那样去理解和制定策略是多么重要。

本书由扎德尔斯基(Zdziarski,

J.)著。

<<iOS应用安全攻防>>

作者简介

作者：（美国）扎德爾斯基（Jonathan Zdziarski）

<<iOS应用安全攻防>>

书籍目录

Preface

1. Everything You Know Is Wrong

The Myth of a Monoculture

The iOS Security Model

Components of the iOS Security Model

Storing the Key with the Lock

Passcodes Equate to Weak Security

Foreic Data Trumps Encryption

External Data Is at Risk, Too

Hijacking Traffic

Data Can Be Stolen...Quickly

Trust No One, Not Even Your Application

Physical Access Is Optional

Summary

Part . Hacking

2. The Basics of Compromising iOS

Why It's Important to Learn How to Break Into a Device

Jailbreaking Explained

Developer Tools

End User Jailbreaks

Jailbreaking an iPhone

DFU Mode

Tethered Veus Untethered

Compromising Devices and Injecting Code

Building Custom Code

Analyzing Your Binary

Testing Your Binary

Daemon!zing Code

Deploying Malicious Code with a Tar Archive

Deploying Malicious Code with a RAM Disk

Exercises

Summary

3. Stealing the Filesystem

Full Disk Encryption

Solid State NAND

Disk Encryption

Where IOS Disk Encryption Has Failed You

Copying the Live Filesystem

The DataTheft Payload

Customizing launchd

Preparing the RAM disk

Imaging the Filesystem

Copying the Raw Filesystem

The RawTheft Payload

Customizing launchd

<<iOS应用安全攻防>>

- Preparing the RAM disk
- Imaging the Filesystem
- Exercises
- The Role of Social Engineering
- Disabled Device Decoy
- Deactivated Device Decoy
- Malware Enabled Decoy
- Password Engineering Application
- Summary

4. Foreic Trace and Data Leakage

- Extracting Image Geotags
- Coolidated GPS Cache
- SQLite Databases
- Connecting to a Database
- SQLite Built-in Commands
- Issuing SQL Queries
- Important Database Files
- Address Book Contacts
- Address Book Images
- Google Maps Data
- Calendar Events
- Call History
- Email Database
- Notes
- Photo Metadata
- SMS Messages
- Safari Bookmarks
- SMS Spotlight Cache
- Safari Web Caches
- Web Application Cache
- WebKit Storage
- Voicemail
- Revee Engineering Remnant Database Fields
- SMS Drafts
- Property Lists
- Important Property List Files
- Other Important Files
- Summary

5. Defeating Encryption

- Sogeti's Data Protection Tools
- Italling Data Protection Tools
- Building the Brute Forcer
- Building Needed Python Libraries
- Extracting Encryption Keys
- The KeyTheft Payload
- Customizing Launchd
- Preparing the RAM disk

<<iOS应用安全攻防>>

- Preparing the Kernel
- Executing the Brute Force
- Decrypting the Keychain
- Decrypting Raw Disk
- Decrypting iTunes Backups
- Defeating Encryption Through Spyware
- The SpyTheft Payload
- Daemonizing spyd
- Customizing Launchd
- Preparing the RAM disk
- Executing the Payload
- Exercises
- Summary

- 6. Unobliterating Files
 - Scraping the HFS Journal
 - Carving Empty Space
 - Commonly Recovered Data
 - Application Screehots
 - Deleted Property Lists
 - Deleted Voicemail and Voice Recordings
 - Deleted Keyboard Cache
 - Photos and Other Peonal Information
 - Summary

- 7. Manipulating the Runtime
 - Analyzing Binaries
 - The Mach-O Format
 - Introduction to class-dump-z
 - Symbol Tables
 - Encrypted Binaries
 - Calculating Offsets
 - Dumping Memory
 - Copy Decrypted Code Back to the File
 - Resetting the cryptid
 - Abusing the Runtime with Cycrypt
 - Italling Cycrypt
 - Using Cycrypt
 - Breaking Simple Locks
 - Replacing Methods
 - Trawling for Data
 - Logging Data
 - More Serious Implicatio
 - Exercises
 - SpringBoard Animatio
 - Call Tapping...Kind Of
 - Making Screen Shots
 - Summary
- 8. Abusingthe Runtime Library

<<iOS应用安全攻防>>

Breaking Objective-C Down

Instance Variables

Methods

Method Cache

Disassembling and Debugging

Eavesdropping

The Underlying Objective-C Framework

Interfacing with Objective-C

Malicious Code Injection

The CodeTheft Payload

Injection Using a Debugger

Injection Using Dynamic Linker Attack

Full Device Infection

Summary

9. Hijacking Traffic

APN Hijacking

Payload Delivery

Removal

Simple Proxy Setup

Attacking SSL

SSLStrip

Paros proxy

Browser Warnings

Attacking Application-Level SSL Validation

The SSLTheft Payload

Hijacking Foundation HTTP Classes

The POSTTheft Payload

Analyzing Data

Driftnet

Building

Running

Exercises

Summary

Part . Securing

10. Implementing Encryption

Password Strength

Beware Random Password Generator

Introduction to Common Crypto

Stateless Operation

Stateful Encryption

Master Key Encryption

Geo-Encryption

Geo-Encryption with Passphrase

Split Server-Side Keys

Securing Memory

Wiping Memory

Public Key Cryptography

<<iOS应用安全攻防>>

Exercises

11. Counter Foreics

- Secure File Wiping
- DOD 5220.22-M Wiping
- Objective-C
- Wiping SQLite Records
- Keyboard Cache
- Randomizing PIN Digits
- Application Screehots

12. Securing the Runtime

- Tamper Respo
- Wipe User Data
- Disable Network Access
- Report Home
- Enable Logging
- False Contacts and Kill Switches
- Process Trace Checking
- Blocking Debugge
- Runtime Class Integrity Checks
- Validating Address Space
- Inline Functio
- Complicating Disassembly
- Optimization Flags
- Stripping
- They're Fun! They Roll! -funroll-loops
- Exercises

13. Jailbreak Detection

- Sandbox Integrity Check
- Filesystem Tests
- Existence of Jailbreak Files
- Size of/etc/fstab
- Evidence of Symbolic Linking
- Page Execution Check

14. Next Steps

- Thinking Like an Attacker
- Other Revee Engineering Tools
- Security Veus Code Management
- A Flexible Approach to Security
- Other Great Books

<<iOS应用安全攻防>>

章节摘录

版权页： 插图： \$ sudo port install cmake libusb-devel libplist Once these packages are installed, create a symlink for the libusb.h prototype so that the usbmuxd package can find it: Download and extract the contents of the usbmuxd source package: Use the cmake command to generate make files, then build the project with make: With usbmuxd installed, and its companion tool iproxy, you'll be able to establish the needed connection bridge from your desktop to your device. While iproxy comes with an open source implementation of usbmuxd, iTunes also includes an officially sanctioned version from Apple that is much faster. To use the much faster version of usbmuxd included with iTunes, ensure that Apple's usbmuxd is loaded and then run the iproxy tool to establish a connection between a local machine (we'll arbitrarily use port 7777 here), and the echo port (port 7) on the device, which is the TCP port your payload code is listening on. \$ sudo launchctl load /System/Library/LaunchDaemons/com.apple.usbmuxd.plist \$ iproxy 7777 7 Once the proxy has started, use netcat (often invoked through its abbreviation nc) to connect to the device through localhost. The netcat utility is a simple tool to make (or listen for) arbitrary network connections, and send or receive data, \$ nc 127.0.0.1 7777 > filesystem.tar The call to nc causes it to connect to the localhost on TCP port 7777. If the proxy and usbmux protocol are working, this connection will be tunneled across USB to the device on port 7, which you specified when you started iproxy. If the connection is working, you should see the device report to the screen that it is sending the/private filesystem, and will see the filesystem.tar file grow on your desktop machine. When the transfer is finished, nc will exit and you will have the complete live user filesystem stored in filesystem.tar! Sometimes, iTunes may not have been properly installed, and you may have problems transferring data from the device. If the tar file remains a zero byte size, try unloading iTunes' copy of usbmuxd and running the open source version you just built.

<<iOS应用安全攻防>>

编辑推荐

《iOS应用安全攻防(影印版)(英文)》由扎德尔斯基著。

检查现实应用中的微小漏洞——并避免在你的应用中出现同样的问题，了解黑客如何通过代码注入来使应用感染恶意软件，明白攻击者如何破解iOS keychain和数据保护加密，使用调试器和定制代码注入来操纵运行时Objective—C环境，阻止攻击者劫持SSL会话和窃取数据流量，安全地删除文件和设计应用来防止数据泄露，避免滥用调试，验证运行时类的健全性，确保你的代码难以跟踪。

<<iOS应用安全攻防>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>