

# 《Android安全攻防权威指南》

## 图书基本信息

书名：《Android安全攻防权威指南》

13位ISBN编号：978711538570X

出版时间：2015-3

作者：[美] Joshua J. Drake,[西] Pau Oliva Fora,[美] Zach Lanier,[美] Collin Mulliner,[美] Stephen A. Ridley,[德] Georg Wincherski

页数：394

译者：诸葛建伟,杨 坤,肖梓航

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu111.com](http://www.tushu111.com)

# 《Android安全攻防权威指南》

## 内容概要

《Android安全攻防权威指南》由世界顶尖级黑客打造，是目前最全面的一本Android系统安全手册。书中细致地介绍了Android系统中的漏洞挖掘、分析，并给出了大量利用工具，结合实例从白帽子角度分析了诸多系统问题，是一本难得的安全指南。移动设备管理者、安全研究员、Android应用程序开发者和负责评估Android安全性的顾问都可以在本书中找到必要的指导和工具。

在本书中你可以：

熟悉实现安全性的细节，以及由Android操作系统开放性带来的复杂问题；

绕开常见安全隐患，了解智能手机最新黑客攻击策略；

回顾曾成功攻破Android操作系统的各类攻击；

探索ROOT操作、分区布局和引导过程；

理解Android生态圈的复杂性，包括各个硬件厂商和软件开发者的影响。

# 《Android安全攻防权威指南》

## 作者简介

Joshua J. Drake

是国际知名黑客，Accuvant LABS公司研究部门总监，曾在世界著名黑客大赛Pwn2Own上攻陷IE浏览器中的Java插件，曾发现Google Glass漏洞。

Pau Oliva Fora

是viaForensics公司的移动安全工程师，为主流Android OEM提供咨询服务。

Zach Lanier

是Duo Security公司的资深安全研究员，在信息安全的不同领域中有十多年的工作经验。

Collin Mulliner

是美国东北大学的博士后研究员，主要研究兴趣是移动和嵌入式系统的安全和隐私，重点关注移动智能手机。

Stephen A. Ridley

是一位安全研究员与技术作者，在软件开发、软件安全和逆向工程领域有十几年的经验。

Georg Wicherski

是CrowdStrike公司的资深安全研究员。

## 书籍目录

|                                  |    |
|----------------------------------|----|
| 第1章 纵观Android生态圈                 | 1  |
| 1.1 了解Android的根源                 | 1  |
| 1.1.1 公司历史                       | 1  |
| 1.1.2 版本历史                       | 2  |
| 1.1.3 审视Android设备家族              | 3  |
| 1.1.4 主体开源                       | 5  |
| 1.2 了解Android的利益相关者              | 6  |
| 1.2.1 谷歌                         | 7  |
| 1.2.2 硬件厂商                       | 7  |
| 1.2.3 移动通信运营商                    | 9  |
| 1.2.4 开发者                        | 9  |
| 1.2.5 用户                         | 10 |
| 1.3 理解生态圈的复杂性                    | 11 |
| 1.3.1 碎片化问题                      | 12 |
| 1.3.2 兼容性                        | 13 |
| 1.3.3 更新问题                       | 13 |
| 1.3.4 安全性与开放性                    | 15 |
| 1.3.5 公开披露                       | 16 |
| 1.4 小结                           | 17 |
| 第2章 Android的安全设计与架构              | 18 |
| 2.1 理解Android系统架构                | 18 |
| 2.2 理解安全边界和安全策略执行                | 19 |
| 2.2.1 Android沙箱                  | 19 |
| 2.2.2 Android权限                  | 22 |
| 2.3 深入理解各个层次                     | 25 |
| 2.3.1 Android应用层                 | 25 |
| 2.3.2 Android框架层                 | 28 |
| 2.3.3 DalvikVM                   | 29 |
| 2.3.4 用户空间原生代码层                  | 30 |
| 2.3.5 内核                         | 36 |
| 2.4 复杂的安全性，复杂的漏洞利用               | 41 |
| 2.5 小结                           | 42 |
| 第3章 root Android设备               | 43 |
| 3.1 理解分区布局                       | 43 |
| 3.2 理解引导过程                       | 45 |
| 3.3 引导加载程序的锁定与解锁                 | 47 |
| 3.4 对未加锁引导加载程序的设备进行root          | 50 |
| 3.5 对锁定引导加载程序的设备进行root           | 52 |
| 3.5.1 在已启动系统中获取root权限            | 52 |
| 3.5.2 NAND 锁、临时性root与永久性root     | 53 |
| 3.5.3 对软root 进行持久化               | 55 |
| 3.6 历史上的一些已知攻击                   | 56 |
| 3.6.1 内核：Wunderbar/asroot        | 56 |
| 3.6.2 恢复：Volez                   | 57 |
| 3.6.3 udev：Exploid               | 57 |
| 3.6.4 adbd：RageAgainstTheCage    | 58 |
| 3.6.5 Zygote：Zimmerlich和Zysploit | 58 |

|  |     |
|--|-----|
| 3.6.6 ashmem : KillingInTheName-Of 和psneuter | 58  |
| 3.6.7 vold : GingerBreak                     | 59  |
| 3.6.8 PowerVR : levitator                    | 59  |
| 3.6.9 libsysutils : zergRush                 | 60  |
| 3.6.10 内核 : mempodroid                       | 60  |
| 3.6.11 文件权限和符号链接相关的攻击                        | 61  |
| 3.6.12 adb 恢复过程竞争条件漏洞                        | 61  |
| 3.6.13 Exynos4 : exynos-abuse                | 62  |
| 3.6.14 Diag : lit/diaggetroot                | 62  |
| 3.7 小结                                       | 63  |
| 第4章 应用安全性评估                                  | 64  |
| 4.1 普遍性安全问题                                  | 64  |
| 4.1.1 应用权限问题                                 | 64  |
| 4.1.2 敏感数据的不安全传输                             | 66  |
| 4.1.3 不安全的数据存储                               | 67  |
| 4.1.4 通过日志的信息泄露                              | 68  |
| 4.1.5 不安全的IPC端点                              | 69  |
| 4.2 案例分析：移动安全应用                              | 71  |
| 4.2.1 初步剖析                                   | 71  |
| 4.2.2 静态分析                                   | 72  |
| 4.2.3 动态分析                                   | 87  |
| 4.2.4 攻击                                     | 95  |
| 4.3 案例分析：SIP客户端                              | 97  |
| 4.3.1 了解Drozer                               | 97  |
| 4.3.2 发现漏洞                                   | 98  |
| 4.3.3 snarfing                               | 99  |
| 4.3.4 注入                                     | 102 |
| 4.4 小结                                       | 104 |
| 第5章 理解Android的攻击面                            | 105 |
| 5.1 攻击基础术语                                   | 105 |
| 5.1.1 攻击向量                                   | 106 |
| 5.1.2 攻击面                                    | 106 |
| 5.2 对攻击面进行分类                                 | 107 |
| 5.2.1 攻击面属性                                  | 108 |
| 5.2.2 分类决策                                   | 108 |
| 5.3 远程攻击面                                    | 108 |
| 5.3.1 网络概念                                   | 109 |
| 5.3.2 网络协议栈                                  | 112 |
| 5.3.3 暴露的网络服务                                | 113 |
| 5.3.4 移动技术                                   | 114 |
| 5.3.5 客户端攻击面                                 | 115 |
| 5.3.6 谷歌的基础设施                                | 119 |
| 5.4 物理相邻                                     | 123 |
| 5.4.1 无线通信                                   | 123 |
| 5.4.2 其他技术                                   | 127 |
| 5.5 本地攻击面                                    | 128 |
| 5.5.1 探索文件系统                                 | 128 |
| 5.5.2 找到其他的本地攻击面                             | 129 |

|                               |     |
|-------------------------------|-----|
| 5.6 物理攻击面                     | 133 |
| 5.6.1 拆解设备                    | 133 |
| 5.6.2 USB                     | 134 |
| 5.6.3 其他物理攻击面                 | 137 |
| 5.7 第三方修改                     | 137 |
| 5.8 小结                        | 137 |
| 第6章 使用模糊测试来挖掘漏洞               | 139 |
| 6.1 模糊测试的背景                   | 139 |
| 6.1.1 选定目标                    | 140 |
| 6.1.2 构造畸形输入                  | 140 |
| 6.1.3 处理输入                    | 141 |
| 6.1.4 监控结果                    | 142 |
| 6.2 Android上的模糊测试             | 142 |
| 6.3 对Broadcast Receiver进行模糊测试 | 143 |
| 6.3.1 选定目标                    | 143 |
| 6.3.2 生成输入                    | 144 |
| 6.3.3 传递输入                    | 145 |
| 6.3.4 监控测试                    | 145 |
| 6.4 对Android上的Chrome进行模糊测试    | 147 |
| 6.4.1 选择一种技术作为目标              | 148 |
| 6.4.2 生成输入                    | 149 |
| 6.4.3 处理输入                    | 151 |
| 6.4.4 监控测试                    | 152 |
| 6.5 对USB攻击面进行模糊测试             | 155 |
| 6.5.1 对USB进行模糊测试的挑战           | 155 |
| 6.5.2 选定目标模式                  | 155 |
| 6.5.3 生成输入                    | 156 |
| 6.5.4 处理输入                    | 158 |
| 6.5.5 监控测试                    | 158 |
| 6.6 小结                        | 159 |
| 第7章 调试与分析安全漏洞                 | 161 |
| 7.1 获取所有信息                    | 161 |
| 7.2 选择一套工具链                   | 162 |
| 7.3 调试崩溃Dump                  | 163 |
| 7.3.1 系统日志                    | 163 |
| 7.3.2 Tombstone               | 164 |
| 7.4 远程调试                      | 165 |
| 7.5 调试Dalvik代码                | 166 |
| 7.5.1 调试示例应用                  | 167 |
| 7.5.2 显示框架层源代码                | 168 |
| 7.5.3 调试现有代码                  | 170 |
| 7.6 调试原生代码                    | 173 |
| 7.6.1 使用NDK进行调试               | 174 |
| 7.6.2 使用Eclipse进行调试           | 177 |
| 7.6.3 使用AOSP进行调试              | 179 |
| 7.6.4 提升自动化程度                 | 183 |
| 7.6.5 使用符号进行调试                | 184 |
| 7.6.6 调试非AOSP设备               | 189 |
| 7.7 调试混合代码                    | 190 |

|                           |     |
|---------------------------|-----|
| 7.8 其他调试技术                | 191 |
| 7.8.1 调试语句                | 191 |
| 7.8.2 在设备上进行调试            | 191 |
| 7.8.3 动态二进制注入             | 192 |
| 7.9 漏洞分析                  | 193 |
| 7.9.1 明确问题根源              | 193 |
| 7.9.2 判断漏洞可利用性            | 205 |
| 7.10 小结                   | 205 |
| 第8章 用户态软件的漏洞利用            | 206 |
| 8.1 内存破坏漏洞基础              | 206 |
| 8.1.1 栈缓冲区溢出              | 206 |
| 8.1.2 堆的漏洞利用              | 209 |
| 8.2 公开的漏洞利用               | 215 |
| 8.2.1 GingerBreak         | 215 |
| 8.2.2 zergRush            | 218 |
| 8.2.3 MempoDroid          | 221 |
| 8.3 Android浏览器漏洞利用        | 222 |
| 8.3.1 理解漏洞                | 222 |
| 8.3.2 控制堆                 | 224 |
| 8.4 小结                    | 227 |
| 第9章 ROP漏洞利用技术             | 228 |
| 9.1 历史和动机                 | 228 |
| 9.2 ARM 架构下的ROP 基础        | 230 |
| 9.2.1 ARM 子函数调用           | 231 |
| 9.2.2 将gadget组成ROP链       | 232 |
| 9.2.3 识别潜在的gadget         | 234 |
| 9.3 案例分析：Android 4.0.1链接器 | 235 |
| 9.3.1 迁移栈指针               | 236 |
| 9.3.2 在新映射内存中执行任意代码       | 237 |
| 9.4 小结                    | 240 |
| 第10章 攻击内核                 | 242 |
| 10.1 Android 的Linux内核     | 242 |
| 10.2 内核提取                 | 242 |
| 10.2.1 从出厂固件中提取内核         | 243 |
| 10.2.2 从设备中提取内核           | 245 |
| 10.2.3 从启动镜像中提取内核         | 246 |
| 10.2.4 解压内核               | 247 |
| 10.3 运行自定义内核代码            | 247 |
| 10.3.1 获取源代码              | 247 |
| 10.3.2 搭建编译环境             | 250 |
| 10.3.3 配置内核               | 251 |
| 10.3.4 使用自定义内核模块          | 252 |
| 10.3.5 编译自定义内核            | 254 |
| 10.3.6 制作引导镜像             | 257 |
| 10.3.7 引导自定义内核            | 258 |
| 10.4 调试内核                 | 262 |
| 10.4.1 获取内核崩溃报告           | 263 |
| 10.4.2 理解Oops信息           | 264 |
| 10.4.3 使用KGDB进行Live调试     | 267 |

|                            |     |
|----------------------------|-----|
| 10.5 内核漏洞利用                | 271 |
| 10.5.1 典型Android内核         | 271 |
| 10.5.2 获取地址                | 273 |
| 10.5.3 案例分析                | 274 |
| 10.6 小结                    | 283 |
| 第11章 攻击RIL无线接口层            | 284 |
| 11.1 RIL简介                 | 284 |
| 11.1.1 RIL架构               | 285 |
| 11.1.2 智能手机架构              | 285 |
| 11.1.3 Android电话栈          | 286 |
| 11.1.4 对电话栈的定制             | 287 |
| 11.1.5 RIL 守护程序            | 287 |
| 11.1.6 用于vendor-ril的API    | 289 |
| 11.2 短信服务                  | 290 |
| 11.2.1 SMS消息的收发            | 290 |
| 11.2.2 SMS消息格式             | 291 |
| 11.3 与调制解调器进行交互            | 293 |
| 11.3.1 模拟调制解调器用于模糊测试       | 293 |
| 11.3.2 在Android中对SMS进行模糊测试 | 295 |
| 11.4 小结                    | 302 |
| 第12章 漏洞利用缓解技术              | 303 |
| 12.1 缓解技术的分类               | 303 |
| 12.2 代码签名                  | 304 |
| 12.3 加固堆缓冲区                | 305 |
| 12.4 防止整数溢出                | 305 |
| 12.5 阻止数据执行                | 306 |
| 12.6 地址空间布局随机化             | 308 |
| 12.7 保护栈                   | 310 |
| 12.8 保护格式化字符串              | 310 |
| 12.9 只读重定位表                | 312 |
| 12.10 沙盒                   | 313 |
| 12.11 增强源代码                | 313 |
| 12.12 访问控制机制               | 315 |
| 12.13 保护内核                 | 316 |
| 12.13.1 指针和日志限制            | 316 |
| 12.13.2 保护零地址页             | 317 |
| 12.13.3 只读的内存区域            | 318 |
| 12.14 其他加固措施               | 318 |
| 12.15 漏洞利用缓解技术总结           | 320 |
| 12.16 禁用缓解机制               | 322 |
| 12.16.1 更改personality      | 322 |
| 12.16.2 修改二进制文件            | 323 |
| 12.16.3 调整内核               | 323 |
| 12.17 对抗缓解技术               | 323 |
| 12.17.1 对抗栈保护              | 324 |
| 12.17.2 对抗ASLR             | 324 |
| 12.17.3 对抗数据执行保护           | 324 |
| 12.17.4 对抗内核级保护机制          | 325 |
| 12.18 展望未来                 | 325 |

|                            |     |
|----------------------------|-----|
| 12.18.1 进行中的官方项目           | 325 |
| 12.18.2 社区的内核加固工作          | 326 |
| 12.18.3 一些预测               | 326 |
| 12.19 小结                   | 327 |
| 第13章 硬件层的攻击                | 328 |
| 13.1 设备的硬件接口               | 328 |
| 13.1.1 UART 串行接口           | 329 |
| 13.1.2 I2C、SPI 和单总线接口      | 331 |
| 13.1.3 JTAG                | 334 |
| 13.1.4 寻找调试接口              | 343 |
| 13.2 识别组件                  | 353 |
| 13.2.1 获得规格说明书             | 353 |
| 13.2.2 难以识别的组件             | 354 |
| 13.3 拦截、监听和劫持数据            | 355 |
| 13.3.1 USB                 | 355 |
| 13.3.2 I2C、SPI和UART串行端口    | 359 |
| 13.4 窃取机密和固件               | 364 |
| 13.4.1 无损地获得固件             | 364 |
| 13.4.2 有损地获取固件             | 365 |
| 13.4.3 拿到dump文件后怎么做        | 368 |
| 13.5 陷阱                    | 371 |
| 13.5.1 定制的接口               | 371 |
| 13.5.2 二进制私有数据格式           | 371 |
| 13.5.3 熔断调试接口              | 372 |
| 13.5.4 芯片密码                | 372 |
| 13.5.5 bootloader密码、热键和哑终端 | 372 |
| 13.5.6 已定制的引导过程            | 373 |
| 13.5.7 未暴露的地址线             | 373 |
| 13.5.8 防止逆向的环氧树脂           | 373 |
| 13.5.9 镜像加密、混淆和反调试         | 373 |
| 13.6 小结                    | 374 |
| 附录A 工具                     | 375 |
| 附录B 开源代码库                  | 386 |

# 《Android安全攻防权威指南》

## 精彩短评

- 1、好啊，牛啊，深啊
- 2、现在看这种书基本看不下去也看不懂。我想了想自己能走的路主要剩下三种：1、社会工程学；2、从所有权角度控制IT公司的核心部分，这是社会工程学的极致形态；3、雇佣一个专业人士。
- 3、读过原版的，不知道中文版翻译的怎么样

# 《Android安全攻防权威指南》

## 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:[www.tushu111.com](http://www.tushu111.com)