

《游戏安全——手游安全技术入门》

图书基本信息

书名：《游戏安全——手游安全技术入门》

13位ISBN编号：9787121287838

出版时间：2016-6

作者：腾讯游戏研发部游戏安全中心

页数：384

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

《游戏安全——手游安全技术入门》

内容概要

《游戏安全——手游安全技术入门》是国内移动游戏安全领域的开山之作，填补了移动游戏安全书籍的空白，揭开了移动游戏外挂的神秘面纱。

随着移动互联网的日益普及，业内对移动安全领域的专业人才的需求逐年增加，而该领域的专业人才相对匮乏，很多开发人员和有志于从事相关行业的在校学生等一直缺少相关的参考资料和书籍。作为移动安全领域的入门书籍，《游戏安全——手游安全技术入门》以移动端（涵盖了Android和iOS两大平台）的游戏逆向分析和外挂技术为切入点，详细讲述了手游安全领域的诸多基础知识和技能，包括：移动端开发和调试环境搭建、典型的移动游戏特性、与外挂相关的安全开发技术、游戏和外挂的逆向分析方法、外挂开发实战演练、游戏引擎逆向分析等内容，书中的部分源代码可免费从网上下载。读者在掌握本书的内容之后，便可入门手游安全领域，同时可以很容易地将在本书中学到的知识扩展至移动端的其他领域，例如：安全方案开发、病毒分析、软件逆向及保护等。

《游戏安全——手游安全技术入门》可作为高等院校计算机安全相关专业的辅助教材，也可供移动端安全技术人员、游戏开发人员，以及有志于从事游戏安全相关工作的学生等参考。

作者简介

邓立丰

在2010年加入腾讯公司，早期从事端游客户端的安全工作，负责过《地下城与勇士》《QQ飞车》《剑灵》等大型PC端网络游戏的反外挂对抗工作；从2014年开始从事手游客户端的安全工作，先后负责过《天天酷跑》《全民飞机大战》《雷霆战机》《天天炫斗》《穿越火线：枪战王者》等移动端游戏的反外挂对抗工作。从事游戏安全行业6年，积累了丰富的游戏安全反外挂对抗经验，对游戏安全领域有较为深入的理解和认知。

陈志豪

在2014年本科毕业于中南大学，以应届毕业生身份进入腾讯公司，担任客户端安全工程师。主要从事腾讯所有自研或代理手游的安全评审、《天天飞车》《天天炫斗》等游戏的外挂对抗，以及其他手游的漏洞挖掘工作。对各种类型的主流游戏的外挂和反外挂原理都有所积累。

王彬

在2014年以应届毕业生身份加入腾讯公司的游戏部门，负责手游前端的安全对抗工作。目前主要负责《全民飞机大战》《王者荣耀》《穿越火线：枪战王者》等手游的反外挂工作，同时开发工具以支持手游安全评审及漏洞挖掘等相关工作。

潘宇峰

毕业于武汉大学，曾任百度安全工程师，在2015年加入腾讯公司，主要负责腾讯公司手游的安全评审和漏洞挖掘工作。熟悉基于Unity 3D、Cocos2d-x引擎的C++、C#、Lua游戏的分析和外挂制作。

刘从刚

在2015年加入腾讯公司的游戏安全部门，参与多个移动端游戏的反外挂对抗和运营工作，平时热爱逆向分析。

彭智伟

在2014年以应届毕业生身份进入腾讯公司，负责手游安全性评审和外挂分析工作。

黎伟柱

在2009年大学毕业，先后从事嵌入式系统和Android软件开发。在2014年加入腾讯公司，从事手游外挂对抗工作。熟悉Android软件框架，擅长软件开发及架构。

丁笑一

在2015年以应届毕业生身份进入腾讯公司的游戏部门，负责手游安全开发和对抗工作。熟悉各类手游的玩法和外挂原理，擅长Android和iOS平台的游戏安全方案的设计及分析工具的开发。

王宏俊

从2013年开始涉足Android安全行业，对Android安全的攻防有一定的经验，曾主导开发了Android版的《叉叉助手》。在2014年加入腾讯公司的业务安全中心，现在主要从事Android客户端安全方案的开发工作。

书籍目录

第 1 篇概述篇 1

第1章手游面临的安全风险 2

1.1 静态修改文件 3

1.1.1 修改游戏资源 3

1.1.2 修改代码 4

1.1.3 修改配置 4

1.2 动态篡改逻辑 4

1.2.1 修改代码 5

1.2.2 修改数据 6

1.3 游戏协议 6

1.3.1 篡改游戏协议 6

1.3.2 重发游戏协议 7

1.4 游戏盗号 7

1.5 恶意发言 8

1.6 工作室 8

1.7 小结 8

第 2 章外挂的定义、分类及实现原理 9

2.1 外挂的定义 9

2.2 外挂的分类 10

2.2.1 辅助版外挂 10

2.2.2 破解版外挂 15

2.3 外挂的实现原理 15

2.3.1 辅助版外挂的实现原理 16

2.3.2 破解版外挂的实现原理 17

2.4 小结 18

第 3 章手游外挂技术汇总 19

3.1 ARM 汇编 19

3.2 C、C++语言 19

3.3 Android 开发 20

3.4 iOS开发 20

3.5 了解常用的游戏引擎 20

3.6 静态分析（IDA 分析） 21

3.7 动态分析（Android、iOS调试） 21

3.8 有必要了解的其他编程语言 21

3.9 静态修改 22

3.10 动态修改 22

3.11 小结 22

第 2 篇环境搭建篇 23

第4章开发环境搭建 24

4.1 Android 开发环境搭建 24

4.1.1 Cygwin 环境搭建 24

4.1.2 Eclipse 环境搭建 27

4.1.3 Android 平台的Native 程序编写 29

4.1.4 Android Native 程序的NDK 编译 30

4.1.5 Android Native 程序的加载运行 30

4.2 iOSXcode开发环境搭建 31

4.2.1 下载Xcode 31

- 4.2.2 真机部署 32
- 4.3 iOS越狱开发环境搭建 33
 - 4.3.1 Theos越狱开发环境搭建 34
 - 4.3.2 iOSOpenDev下载与安装 35
 - 4.3.3 如何创建和编译iOS动态库文件 36
 - 4.3.4 如何加载、运行iOS动态库 37
- 4.4 小结 38
- 第5章 调试环境搭建 39
 - 5.1 Android 平台调试环境的搭建 39
 - 5.2 iOS 32 位调试环境的搭建 41
 - 5.2.1 软件安装 41
 - 5.2.2 iOS 32 位程序的调试 42
 - 5.3 iOS 64 位程序调试环境的搭建 44
 - 5.3.1 iPhone 设备的CPU 类型介绍 44
 - 5.3.2 lldb环境搭建 45
 - 5.3.3 lldb调试介绍 46
 - 5.4 小结 48
- 第6章 工具汇总与使用 49
 - 6.1 IDA Pro 49
 - 6.1.1 用IDA 加载可执行文件 50
 - 6.1.2 用IDA 分析可执行文件 52
 - 6.1.3 IDA 功能界面 54
 - 6.2 APKTool工具 61
 - 6.2.1 反编译APK 文件 62
 - 6.2.2 重打包APK 文件 63
 - 6.3 ILSpy工具 64
 - 6.3.1 加载文件 64
 - 6.3.2 保存反编译代码 65
 - 6.4 MachOView工具 66
 - 6.4.1 加载Mach-O 文件 67
 - 6.4.2 文件头信息 68
 - 6.4.3 加密信息获取 69
 - 6.5 MobileSubStrate工具组件 70
 - 6.5.1 MobileHooker 71
 - 6.5.2 MobileLoader 71
 - 6.5.3 Safe Mode 72
 - 6.6 小结 72
- 第3篇 游戏基础篇 73
- 第7章 手游开发基础概述 74
 - 7.1 游戏玩法与分类 74
 - 7.1.1 MMORPG 类游戏 75
 - 7.1.2 FPS 类游戏 77
 - 7.1.3 ARPG 类游戏 78
 - 7.1.4 卡牌类游戏 79
 - 7.1.5 RTS 类游戏 79
 - 7.1.6 消除类游戏 80
 - 7.1.7 MOBA 类游戏 81
 - 7.1.8 跑酷类游戏 81
 - 7.2 游戏系统及开发的相关概念 82

- 7.2.1 手游系统的组成 82
- 7.2.2 手游开发语言 88
- 7.2.3 手游网络模式 88
- 7.3 小结 89
- 第8章 游戏引擎的基本概念及常见引擎介绍 90
- 8.1 什么是游戏引擎 90
- 8.2 游戏引擎子系统 91
 - 8.2.1 渲染系统 91
 - 8.2.2 音频系统 92
 - 8.2.3 物理系统 93
 - 8.2.4 人工智能 93
- 8.3 常用手游引擎 94
 - 8.3.1 Cocos2D 引擎 94
 - 8.3.2 Unity 3D 引擎 95
- 8.4 小结 96
- 第9章 游戏漏洞概述 97
- 9.1 游戏安全漏洞的基本概念 97
 - 9.1.1 游戏逻辑漏洞 98
 - 9.1.2 游戏协议稳定型漏洞 98
 - 9.1.3 游戏服务端校验疏忽型漏洞 99
- 9.2 游戏漏洞风险点分类 99
 - 9.2.1 手游常见类型 99
 - 9.2.2 手游风险 100
- 9.3 小结 104
- 第4篇 逆向篇 105
- 第10章 静态分析 106
- 10.1 ARM 反汇编速成 106
 - 10.1.1 ARM 体系简介 106
 - 10.1.2 ARM 指令样例解析 107
 - 10.1.3 Thumb 指令简述 110
 - 10.1.4 函数传参 111
 - 10.1.5 浮点数基础 111
- 10.2 Android 平台的ELF 文件格式 113
 - 10.2.1 文件头信息 114
 - 10.2.2 程序头信息 115
 - 10.2.3 节表头信息 117
- 10.3 iOS平台的Mach-O 文件格式 118
 - 10.3.1 文件头格式 119
 - 10.3.2 Load Command 信息 121
- 10.4 IDA 静态分析 123
 - 10.4.1 IDA 启动及加载文件 123
 - 10.4.2 IDA 静态分析主界面及窗口 124
 - 10.4.3 用IDA 保存静态分析结果 128
 - 10.4.4 IDA 静态分析的常用功能及快捷键 129
- 10.5 小结 132
- 第11章 动态分析 133
- 11.1 Android 平台的IDA 动态调试 133
 - 11.1.1 启动IDA 调试器 133
 - 11.1.2 加载Android 原生动态链接库 135

- 11.1.3 动态调试主界面 138
- 11.1.4 IDA 动态调试断点和脚本功能 139
- 11.1.5 IDA 动态调试修改数据功能 141
- 11.1.6 用IDA 调试器修改代码 143
- 11.2 iOS平台中的GDB 动态调试 144
 - 11.2.1 用GDB 加载调试程序 144
 - 11.2.2 GDB 常用的调试功能 146
- 11.3 iOS平台的lldb动态调试 151
 - 11.3.1 用lldb加载调试程序 151
 - 11.3.2 lldb的调试功能 154
 - 11.3.3 其他功能 157
- 11.4 小结 158
- 第5篇开发篇 159
- 第12章定制化外挂开发流程 160
 - 12.1 什么是定制化外挂 160
 - 12.2 定制化外挂开发的基础流程 161
 - 12.3 定制化外挂开发各环节介绍 161
 - 12.3.1 逆向分析游戏逻辑 162
 - 12.3.2 验证外挂功能是否可行 162
 - 12.3.3 注入游戏进程 163
 - 12.3.4 枚举游戏进程模块 163
 - 12.3.5 Hook 关键函数 163
 - 12.3.6 游戏内存数据修改 164
 - 12.3.7 反调试功能 164
 - 12.4 小结 165
- 第13章注入技术的实现原理 166
 - 13.1 什么是进程注入技术 166
 - 13.2 Android 平台下ptrace注入技术的实现 167
 - 13.2.1 ptrace函数介绍 167
 - 13.2.2 ptrace注入进程流程 168
 - 13.2.3 ptrace注入的实现 169
 - 13.2.4 ptrace注入实例测试 173
 - 13.3 Android 平台下Zygote 注入技术的实现 174
 - 13.3.1 Zygote 注入技术的原理 174
 - 13.3.2 Zygote 注入技术的实现流程 174
 - 13.3.3 Zygote 注入器的实现方式 175
 - 13.3.4 注入Zygote 的模块功能实现 182
 - 13.3.5 Zygote 注入实例测试 182
 - 13.4 Android 平台感染ELF 文件的注入技术实现 184
 - 13.4.1 ELF 文件的格式 185
 - 13.4.2 感染ELF 文件的注入实现原理 186
 - 13.4.3 感染ELF 文件的注入实现过程 187
 - 13.4.4 感染ELF 文件的注入实例分析 188
 - 13.4.5 感染ELF 文件的注入编程实现 192
 - 13.4.6 感染ELF 文件的注入实例测试 194
 - 13.5 iOS平台越狱环境的注入实现 195
 - 13.5.1 利用Theos环境创建注入工程 195
 - 13.5.2 工程文件说明 196
 - 13.5.3 编译和安装 200

- 13.5.4 iOS注入原理介绍 202
- 13.6 小结 203
- 第 14 章 Hook 技术的实现原理 205
 - 14.1 Hook 技术简介 205
 - 14.2 Android 平台基于异常的Hook 实现 206
 - 14.2.1 基于异常Hook 的实现原理 206
 - 14.2.2 Android 平台基于异常Hook 的实现流程 207
 - 14.2.3 基于异常Hook 的实现代码 208
 - 14.2.4 基于异常Hook 的实例测试 212
 - 14.3 Android 平台的Inline Hook 实现 214
 - 14.3.1 Inline Hook 的实现原理 214
 - 14.3.2 Inline Hook 的实现流程 215
 - 14.3.3 Inline Hook 的实现代码 216
 - 14.3.4 Inline Hook 的实例测试 220
 - 14.4 Android 平台下导入表Hook 的实现 224
 - 14.4.1 导入表Hook 的实现原理 224
 - 14.4.2 导入表Hook 的实现流程 224
 - 14.4.3 导入表Hook 的实现代码 225
 - 14.4.4 Android 平台下导入表Hook 的实例测试 229
 - 14.5 小结 230
- 第 15 章 游戏进程的模块信息获取 231
 - 15.1 Android 平台进程模块的信息获取 231
 - 15.1.1 Android 内存模块遍历的原理 232
 - 15.1.2 Android 内存模块遍历的实现 233
 - 15.1.3 实例测试 236
 - 15.2 iOS平台进程模块信息的获取 237
 - 15.2.1 Dyld API 遍历模块的原理 237
 - 15.2.2 Dyld API 遍历模块实现 239
 - 15.2.3 通过内存遍历法获取模块的原理 239
 - 15.2.4 通过内存遍历法获取模块信息的实现 241
 - 15.2.5 实例测试 247
 - 15.3 小结 249
- 第 16 章 篡改游戏内容的实现原理 250
 - 16.1 游戏内容读写方式分类 250
 - 16.2 非注入式篡改 251
 - 16.2.1 篡改APK 安装包 251
 - 16.2.2 篡改游戏的安装目录文件 253
 - 16.2.3 篡改“/proc/”目录文件 253
 - 16.3 注入式篡改 258
 - 16.3.1 篡改内存数据 259
 - 16.3.2 篡改逻辑代码 259
 - 16.3.3 注入式篡改代码实例讲解 260
 - 16.4 小结 263
- 第 17 章 反调试技术 264
 - 17.1 Android 平台的常规反调试技术 264
 - 17.1.1 Android 平台的Self-Debugging 反调试方案 265
 - 17.1.2 Android 平台的轮询检测反调试方案 268
 - 17.1.3 Android 平台Java 层的反调试技术 270
 - 17.2 iOS平台的反调试技术 270

- 17.2.1 iOS平台拒绝被附加反调试方案 271
- 17.2.2 iOS平台的轮询检测反调试方案 272
- 17.3 小结 273
- 第6篇 实战篇 275
- 第18章 游戏逆向分析实战篇 276
- 18.1 C++游戏分析实战篇 276
- 18.1.1 C++游戏识别 277
- 18.1.2 C++基础 278
- 18.1.3 C++游戏的逆向分析方法 279
- 18.1.4 C++游戏的破解思路 281
- 18.1.5 C++游戏逆向分析实战篇——《雷霆战机》无敌和秒杀功能分析 281
- 18.2 Unity 3D 游戏分析实战篇 289
- 18.2.1 识别Unity 3D 游戏 289
- 18.2.2 Unity 3D 游戏的破解方法 290
- 18.2.3 Unity 3D 游戏分析涉及的工具 291
- 18.2.4 Unity 3D 游戏分析实战篇——《星河战神》的无限冲刺功能分析 292
- 18.3 Lua游戏分析实战 295
- 18.3.1 识别Lua游戏 295
- 18.3.2 破解Lua游戏的方法 296
- 18.3.3 常用工具 298
- 18.3.4 Lua游戏实战——破解《疾风猎人》的Lua代码 298
- 18.3.5 Lua游戏分析实战——破解《游龙英雄》的Lua代码 300
- 18.4 小结 302
- 第19章 外挂逆向分析实战——《雷霆战机》圈圈外挂分析 303
- 19.1 外挂整体分析 303
- 19.2 外挂注入功能分析 304
- 19.2.1 com.oozhushou-1.apk 文件分析 304
- 19.2.2 hhloader模块分析 309
- 19.2.3 外挂注入的实现方式 311
- 19.3 外挂作弊功能分析 312
- 19.3.1 秒杀功能的实现分析 312
- 19.3.2 忽略伤害功能的实现分析 315
- 19.3.3 护盾延迟功能的实现分析 316
- 19.4 小结 317
- 第20章 外挂开发实战——《2048》手游快速通关功能分析及开发 318
- 20.1 游戏功能分析 318
- 20.1.1 功能可行性分析 318
- 20.1.2 游戏引擎的确认 320
- 20.1.3 关键逻辑的逆向分析 320
- 20.2 外挂功能的实现 326
- 20.2.1 实现思路 326
- 20.2.2 实现原理 327
- 20.2.3 实现流程 327
- 20.2.4 实现代码 328
- 20.3 测试结果 329
- 20.4 小结 330
- 第21章 Unity 3D 引擎逆向分析 332
- 21.1 Unity 3D 引擎概述 332
- 21.2 Android 平台Unity 3D 引擎的Mono 机制分析 333

- 21.2.1 Mono 介绍 333
- 21.2.2 Mono 主框架的执行流程 334
- 21.2.3 C#函数调用的执行过程 335
- 21.3 iOS平台的Unity 3D 引擎IL2CPP 机制分析 338
 - 21.3.1 IL2CPP 机制生成代码的对比 338
 - 21.3.2 逆向分析IL2CPP 机制中C#函数的调用方式 342
 - 21.3.3 Unity 3D 引擎的IL2CPP 机制安全性分析 347
- 21.4 Unity 3D 引擎AB 机制分析及《天天来战》AB 包还原 349
 - 21.4.1 Unity 3D 的AB 打包实现 349
 - 21.4.2 C#脚本的AB 包加载及运行过程 351
 - 21.4.3 《天天来战》游戏的AB 包处理方式分析及还原 352
- 21.5 小结 358

精彩短评

1、虽然有些内容太专，但用到的时候翻翻还是挺高效的

安全风险：静态修改文件；动态修改代码；修改协议；游戏盗号；恶意发言；工作室；

外挂：辅助类；破解版

纠正了我arpg游戏的错误认识，动作游戏+角色扮演

2、手游目前已经逐渐成为一种潮流，基本上每个人都会玩，所以手游产业也是日益壮大。安全技术也开始受到了重视，不过这块的话核心还是在一些大牛手中，这次是无意中搜到这本书，买完以后读了一下，推荐给了研发的同事都说拿来培训新人不错，所以准备再多买几本。期待有更深入的教学书籍。

3、内容还算比较细致，对于新手可以了解一下

4、我买这本书的原因是这是腾讯gad出的，相信腾讯也相信gad这个教育平台，出的这几本书让我们这些做游戏的新人读后受益匪浅。

5、这本书真的给我这种刚入门的人提供了很大帮助，看了前一部分就懂了好多，继续看~

6、买了，看了，虽然个人对游戏安全了解不深，但是看了这本书，了解了不少游戏安全。并且帮我解决了不少游戏安全问题。力荐！

7、我看书有个习惯，拿到书之后，我会先看目录，以便了解到整本书的构架，其中有部分讲述了如何处理外挂等，还有外挂的实战演练。

8、本书对于游戏安全方面讲解得很详细，覆盖面很广哦~让我更加了解移动游戏外挂

9、很适合完全没有经验的初学者，懂得外挂常见的平常处理方法~

10、差评！

11、买的时候是同一系列的就一起买了，没想到每一本内容都很到点，力赞！

12、从目录到内容，真的讲得很详细，而且调理很清晰，另外两本我也都买了，完全值得

13、还不错。

14、这本书写的很详细，我觉得对入门很有帮助

15、从外挂等角度入手，给人不错的感觉。还是很适合入门级的学习的。

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