# Contents

<table>
<thead>
<tr>
<th>Introduction</th>
<th>xxv</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chapter 1  Looking at the Ecosystem</strong></td>
<td>1</td>
</tr>
<tr>
<td>Understanding Android's Roots</td>
<td>1</td>
</tr>
<tr>
<td>Company History</td>
<td>2</td>
</tr>
<tr>
<td>Version History</td>
<td>2</td>
</tr>
<tr>
<td>Examining the Device Pool</td>
<td>4</td>
</tr>
<tr>
<td>Open Source, Mostly</td>
<td>7</td>
</tr>
<tr>
<td>Understanding Android Stakeholders</td>
<td>7</td>
</tr>
<tr>
<td>Google</td>
<td>8</td>
</tr>
<tr>
<td>Hardware Vendors</td>
<td>10</td>
</tr>
<tr>
<td>Carriers</td>
<td>12</td>
</tr>
<tr>
<td>Developers</td>
<td>13</td>
</tr>
<tr>
<td>Users</td>
<td>14</td>
</tr>
<tr>
<td>Grasping Ecosystem Complexities</td>
<td>15</td>
</tr>
<tr>
<td>Fragmentation</td>
<td>16</td>
</tr>
<tr>
<td>Compatibility</td>
<td>17</td>
</tr>
<tr>
<td>Update Issues</td>
<td>18</td>
</tr>
<tr>
<td>Security versus Openness</td>
<td>21</td>
</tr>
<tr>
<td>Public Disclosures</td>
<td>22</td>
</tr>
<tr>
<td>Summary</td>
<td>23</td>
</tr>
<tr>
<td><strong>Chapter 2 Android Security Design and Architecture</strong></td>
<td>25</td>
</tr>
<tr>
<td>Understanding Android System Architecture</td>
<td>25</td>
</tr>
<tr>
<td>Understanding Security Boundaries and Enforcement</td>
<td>27</td>
</tr>
<tr>
<td>Android's Sandbox</td>
<td>27</td>
</tr>
<tr>
<td>Android Permissions</td>
<td>30</td>
</tr>
<tr>
<td>Looking Closer at the Layers</td>
<td>34</td>
</tr>
<tr>
<td>Android Applications</td>
<td>34</td>
</tr>
<tr>
<td>The Android Framework</td>
<td>39</td>
</tr>
</tbody>
</table>
Chapter 3  Rooting Your Device  57
Understanding the Partition Layout  58
  Determining the Partition Layout  59
Understanding the Boot Process  60
  Accessing Download Mode  61
Locked and Unlocked Boot Loaders  62
  Stock and Custom Recovery Images  63
Rooting with an Unlocked Boot Loader  65
Rooting with a Locked Boot Loader  68
  Gaining Root on a Booted System  69
  NAND Locks, Temporary Root, and Permanent Root  70
  Persisting a Soft Root  71
History of Known Attacks  73
  Kernel: Wunderbar/asroot  73
  Recovery: Volez  74
  Udev: Exploid  74
  Adbd: RageAgainstTheCage  75
  Zygote: Zimperlich and Zysploit  75
  Ashmem: KillingInTheNameOf and psneuter  76
  Vold: GingerBreak  76
  PowerVR: levitator  77
  Libsysutils: zergRush  78
  Kernel: mempodroid  78
  File Permission and Symbolic Link–Related Attacks  79
  Adb Restore Race Condition  79
  Exynos4: exynos-abuse  80
  Diag: lit / diaggetroot  81
Summary  81

Chapter 4  Reviewing Application Security  83
Common Issues  83
  App Permission Issues  84
  Insecure Transmission of Sensitive Data  86
  Insecure Data Storage  87
  Information Leakage Through Logs  88
  Unsecured IPC Endpoints  89
Case Study: Mobile Security App  91
  Profiling  91
  Static Analysis  93
  Dynamic Analysis  109
  Attack  117
## Case Study: SIP Client 120
- Enter Drozer 121
- Discovery 121
- Snarling 122
- Injection 124

## Chapter 5: Understanding Android’s Attack Surface 129
- An Attack Terminology Primer 130
  - Attack Vectors 130
  - Attack Surfaces 131
- Classifying Attack Surfaces 133
  - Surface Properties 133
  - Classification Decisions 134
- Remote Attack Surfaces 134
  - Networking Concepts 134
  - Networking Stacks 139
  - Exposed Network Services 140
  - Mobile Technologies 142
  - Client-side Attack Surface 143
  - Google Infrastructure 148
- Physical Adjacency 154
  - Wireless Communications 154
  - Other Technologies 161
- Local Attack Surfaces 161
  - Exploring the File System 162
  - Finding Other Local Attack Surfaces 163
- Physical Attack Surfaces 168
  - Dismantling Devices 169
  - USB 169
  - Other Physical Attack Surfaces 173
- Third-Party Modifications 174

## Chapter 6: Finding Vulnerabilities with Fuzz Testing 177
- Fuzzing Background 177
  - Identifying a Target 179
  - Crafting Malformed Inputs 179
  - Processing Inputs 180
  - Monitoring Results 181
- Fuzzing on Android 181
- Fuzzing Broadcast Receivers 183
  - Identifying a Target 183
  - Generating Inputs 184
  - Delivering Inputs 185
  - Monitoring Testing 185
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzing Chrome for Android</td>
<td>188</td>
</tr>
<tr>
<td>Selecting a Technology to Target</td>
<td>188</td>
</tr>
<tr>
<td>Generating Inputs</td>
<td>190</td>
</tr>
<tr>
<td>Processing Inputs</td>
<td>192</td>
</tr>
<tr>
<td>Monitoring Testing</td>
<td>194</td>
</tr>
<tr>
<td>Fuzzing the USB Attack Surface</td>
<td>197</td>
</tr>
<tr>
<td>USB Fuzzing Challenges</td>
<td>198</td>
</tr>
<tr>
<td>Selecting a Target Mode</td>
<td>198</td>
</tr>
<tr>
<td>Generating Inputs</td>
<td>199</td>
</tr>
<tr>
<td>Processing Inputs</td>
<td>201</td>
</tr>
<tr>
<td>Monitoring Testing</td>
<td>202</td>
</tr>
<tr>
<td>Summary</td>
<td>204</td>
</tr>
</tbody>
</table>

**Chapter 7  Debugging and Analyzing Vulnerabilities**  

Getting All Available Information                               205  
Choosing a Toolchain                                            207  
Debugging with Crash Dumps                                      208  
  System Logs                                                    208  
  Tombstones                                                     209  
Remote Debugging                                                211  
Debugging Dalvik Code                                           212  
  Debugging an Example App                                      213  
  Showing Framework Source Code                                  215  
  Debugging Existing Code                                       217  
Debugging Native Code                                           221  
  Debugging with the NDK                                        222  
  Debugging with Eclipse                                        226  
  Debugging with AOSP                                           227  
  Increasing Automation                                         233  
  Debugging with Symbols                                        235  
  Debugging with a Non-AOSP Device                              241  
Debugging Mixed Code                                            243  
Alternative Debugging Techniques                                 243  
  Debug Statements                                              243  
  On-Device Debugging                                           244  
  Dynamic Binary Instrumentation                                245  
Vulnerability Analysis                                          246  
  Determining Root Cause                                        246  
  Judging Exploitability                                        260  
Summary                                                          261  

**Chapter 8  Exploiting User Space Software**  

Memory Corruption Basics                                         263  
  Stack Buffer Overflows                                         264  
  Heap Exploitation                                              268  

### Chapter 11  Attacking the Radio Interface Layer  

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to the RIL</td>
<td>368</td>
</tr>
<tr>
<td>RIL Architecture</td>
<td>368</td>
</tr>
<tr>
<td>Smartphone Architecture</td>
<td>369</td>
</tr>
<tr>
<td>The Android Telephony Stack</td>
<td>370</td>
</tr>
<tr>
<td>Telephony Stack Customization</td>
<td>371</td>
</tr>
<tr>
<td>The RIL Daemon (rild)</td>
<td>372</td>
</tr>
<tr>
<td>The Vendor-RIL API</td>
<td>374</td>
</tr>
<tr>
<td>Short Message Service (SMS)</td>
<td>375</td>
</tr>
<tr>
<td>Sending and Receiving SMS Messages</td>
<td>376</td>
</tr>
<tr>
<td>SMS Message Format</td>
<td>376</td>
</tr>
<tr>
<td>Interacting with the Modem</td>
<td>379</td>
</tr>
<tr>
<td>Emulating the Modem for Fuzzing</td>
<td>379</td>
</tr>
<tr>
<td>Fuzzing SMS on Android</td>
<td>382</td>
</tr>
<tr>
<td>Summary</td>
<td>390</td>
</tr>
</tbody>
</table>

### Chapter 12  Exploit Mitigations  

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classifying Mitigations</td>
<td>392</td>
</tr>
<tr>
<td>Code Signing</td>
<td>392</td>
</tr>
<tr>
<td>Hardening the Heap</td>
<td>394</td>
</tr>
<tr>
<td>Protecting Against Integer Overflows</td>
<td>394</td>
</tr>
<tr>
<td>Preventing Data Execution</td>
<td>396</td>
</tr>
<tr>
<td>Address Space Layout Randomization</td>
<td>398</td>
</tr>
<tr>
<td>Protecting the Stack</td>
<td>400</td>
</tr>
<tr>
<td>Format String Protections</td>
<td>401</td>
</tr>
<tr>
<td>Read-Only Relocations</td>
<td>403</td>
</tr>
<tr>
<td>Sandboxing</td>
<td>404</td>
</tr>
<tr>
<td>Fortifying Source Code</td>
<td>405</td>
</tr>
<tr>
<td>Access Control Mechanisms</td>
<td>407</td>
</tr>
<tr>
<td>Protecting the Kernel</td>
<td>408</td>
</tr>
<tr>
<td>Pointer and Log Restrictions</td>
<td>409</td>
</tr>
<tr>
<td>Protecting the Zero Page</td>
<td>410</td>
</tr>
<tr>
<td>Read-Only Memory Regions</td>
<td>410</td>
</tr>
<tr>
<td>Other Hardening Measures</td>
<td>411</td>
</tr>
<tr>
<td>Summary of Exploit Mitigations</td>
<td>414</td>
</tr>
<tr>
<td>Disabling Mitigation Features</td>
<td>415</td>
</tr>
<tr>
<td>Changing Your Personality</td>
<td>416</td>
</tr>
<tr>
<td>Altering Binaries</td>
<td>416</td>
</tr>
<tr>
<td>Tweaking the Kernel</td>
<td>417</td>
</tr>
<tr>
<td>Overcoming Exploit Mitigations</td>
<td>418</td>
</tr>
<tr>
<td>Overcoming Stack Protections</td>
<td>418</td>
</tr>
<tr>
<td>Overcoming ASLR</td>
<td>418</td>
</tr>
<tr>
<td>Overcoming Data Execution Protections</td>
<td>419</td>
</tr>
<tr>
<td>Overcoming Kernel Protections</td>
<td>419</td>
</tr>
</tbody>
</table>
Contents

Looking to the Future 420
   Official Projects Underway 420
   Community Kernel Hardening Efforts 420
   A Bit of Speculation 422
Summary 422

Chapter 13 Hardware Attacks 423
Interfacing with Hardware Devices 424
   UART Serial Interfaces 424
   I2C, SPI, and One-Wire Interfaces 428
   JTAG 431
      Finding Debug Interfaces 443
Identifying Components 456
   Getting Specifications 456
   Difficulty Identifying Components 457
Intercepting, Monitoring, and Injecting Data 459
   USB 459
      I2C, SPI, and UART Serial Interfaces 463
Stealing Secrets and Firmware 469
   Accessing Firmware Unobtrusively 469
   Destructively Accessing the Firmware 471
   What Do You Do with a Dump? 474
Pitfalls 479
   Custom Interfaces 479
   Binary/Proprietary Data 479
   Blown Debug Interfaces 480
   Chip Passwords 480
   Boot Loader Passwords, Hotkeys, and Silent Terminals 480
   Customized Boot Sequences 481
   Unexposed Address Lines 481
   Anti-Reversing Epoxy 482
   Image Encryption, Obfuscation, and Anti-Debugging 482
Summary 482

Appendix A Tool Catalog 485
Development Tools 485
   Android SDK 485
   Android NDK 486
   Eclipse 486
   ADT Plug-In 486
   ADT Bundle 486
   Android Studio 487
Firmware Extraction and Flashing Tools 487
   Binwalk 487
   fastboot 487
Samsung 488
NVIDIA 489
LG 489
HTC 489
Motorola 490
Native Android Tools 491
BusyBox 491
setpropex 491
SQLite 491
strace 492
Hooking and Instrumentation Tools 492
ADB I Framework 492
ldpreloadhook 492
XPosed Framework 492
Cydia Substrate 493
Static Analysis Tools 493
Small and Baksmali 493
Androguard 493
apktool 494
dex2jar 494
jad 494
JD-GUI 495
JEB 495
Radare2 495
IDA Pro and Hex-Rays Decompiler 496
Application Testing Tools 496
Drozer (Mercury) Framework 496
iSEC Intent Sniffer and Intent Fuzzer 496
Hardware Hacking Tools 496
Segger J-Link 497
JTAGulator 497
OpenOCD 497
Saleae 497
Bus Pirate 497
GoodFET 497
Total Phase Beagle USB 498
Facedancer21 498
Total Phase Beagle I2C 498
Chip Quik 498
Hot air gun 498
Xeltek SuperPro 498
IDA 499

Appendix B Open Source Repositories
Google 501
AOSP 501
Gerrit Code Review 502
SoC Manufacturers 502
   AllWinner 503
   Intel 503
   Marvell 503
   MediaTek 504
   Nvidia 504
   Texas Instruments 504
   Qualcomm 505
   Samsung 505
OEMs 506
   ASUS 506
   HTC 507
   LG 507
   Motorola 507
   Samsung 508
   Sony Mobile 508
Upstream Sources 508
Others 509
   Custom Firmware 509
   Linaro 510
   Replicant 510
   Code Indexes 510
   Individuals 510

Appendix C References 511

Index 523