CONTENTS IN DETAIL

FOREWORD **xix**

PREFACE **xxi**

ACKNOWLEDGMENTS **xxv**

INTRODUCTION **xxvii**

Who Should Read This Book? xxviii
What Topics Does the Book Cover? xxix
    Part I: Concepts xxix
    Part II: Design xxix
    Part III: Implementation xxx
    Conclusion xxx
    Appendices xxxi
Good, Safe Fun xxxi

PART I: CONCEPTS 1

1
FOUNDATIONS **3**

Understanding Security 4
Trust 5
    Feeling Trust 6
    You Cannot See Bits 6
    Competence and Imperfection 7
    Trust Is a Spectrum 8
    Trust Decisions 8
    Implicitly Trusted Components 9
    Being Trustworthy 10
Classic Principles 10
    Information Security's C-I-A 11
    The Gold Standard 14
    Privacy 19

2
THREATS **23**

The Adversarial Perspective 24
The Four Questions 25
Threat Modeling 26
    Work from a Model 27
    Identify Assets 28
    Identify Attack Surfaces 30

    Identify Trust Boundaries 30
    Identify Threats 33
    Mitigate Threats 38
Privacy Considerations 39
Threat Modeling Everywhere 40

3
MITIGATION 43

Addressing Threats 44
Structural Mitigation Strategies 45
    Minimize Attack Surfaces 45
    Narrow Windows of Vulnerability 46
    Minimize Data Exposure 47
Access Policy and Access Controls 48
Interfaces 49
Communication 50
Storage 51

4
PATTERNS 53

Design Attributes 54
    Economy of Design 54
    Transparent Design 56
Exposure Minimization 56
    Least Privilege 56
    Least Information 57
    Secure by Default 59
    Allowlists over Blocklists 60
    Avoid Predictability 61
    Fail Securely 62
Strong Enforcement 62
    Complete Mediation 63
    Least Common Mechanism 64
Redundancy 65
    Defense in Depth 65
    Separation of Privilege 67
Trust and Responsibility 68
    Reluctance to Trust 68
    Accept Security Responsibility 69
Anti-Patterns 71
    Confused Deputy 71
    Backflow of Trust 73
    Third-Party Hooks 74
    Unpatchable Components 74

5
CRYPTOGRAPHY 75

Crypto Tools 76
Random Numbers 77
    Pseudo-Random Numbers 77
    Cryptographically Secure Pseudo-Random Numbers 77

Message Authentication Codes . . . 78
    Using MACs to Prevent Tampering . . . 79
    Replay Attacks . . . 79
    Secure MAC Communications . . . 80
Symmetric Encryption . . . 81
    One-Time Pad . . . 81
    Advanced Encryption Standard . . . 82
    Using Symmetric Cryptography . . . 83
Asymmetric Encryption . . . 83
    The RSA Cryptosystem . . . 84
Digital Signatures . . . 85
Digital Certificates . . . 86
Key Exchange . . . 87
Using Crypto . . . 89

PART II: DESIGN 93

6
SECURE DESIGN 95

Integrating Security in Design . . . 96
    Making Design Assumptions Explicit . . . 97
    Defining the Scope . . . 98
    Setting Security Requirements . . . 99
    Threat Modeling . . . 101
Building in Mitigations . . . 103
    Designing Interfaces . . . 103
    Designing Data Handling . . . 104
Integrating Privacy into Design . . . 105
Planning for the Full Software Lifecycle . . . 106
Making Trade-Offs . . . 106
Design Simplicity . . . 107

7
SECURITY DESIGN REVIEWS 109

SDR Logistics . . . 110
    Why Conduct an SDR? . . . 110
    When to Conduct an SDR . . . 110
    Documentation Is Essential . . . 111
The SDR Process . . . 111
    1. Study . . . 112
    2. Inquire . . . 112
    3. Identify . . . 113
    4. Collaborate . . . 113
    5. Write . . . 114
    6. Follow Up . . . 116
Assessing Design Security . . . 116
    Using the Four Questions as Guidance . . . 116
    Where to Dig . . . 119
    Privacy Reviews . . . 120
    Reviewing Updates . . . 120

Managing Disagreement . . . 121
    Communicate Tactfully . . . 121
    Case Study: A Difficult Review . . . 122
    Escalating Disagreements . . . 123
Practice, Practice, Practice . . . 124

PART III: IMPLEMENTATION 127

8

SECURE PROGRAMMING 129

The Challenge . . . 130
    Malicious Influence . . . 131
    Vulnerabilities Are Bugs . . . 133
    Vulnerability Chains . . . 134
    Bugs and Entropy . . . 135
    Vigilance . . . 136
Case Study: GotoFail . . . 137
    One-Line Vulnerability . . . 137
    Beware of Footguns . . . 138
    Lessons from GotoFail . . . 139
Coding Vulnerabilities . . . 140
    Atomicity . . . 140
    Timing Attacks . . . 141
    Serialization . . . 142
The Usual Suspects . . . 143

9

LOW-LEVEL CODING FLAWS 145

Arithmetic Vulnerabilities . . . 146
    Fixed-Width Integer Vulnerabilities . . . 147
    Floating-Point Precision Vulnerabilities . . . 150
    Example: Floating-Point Underflow . . . 151
    Example: Integer Overflow . . . 153
    Safe Arithmetic . . . 155
Memory Access Vulnerabilities . . . 156
    Memory Management . . . 157
    Buffer Overflow . . . 157
    Example: Memory Allocation Vulnerabilities . . . 158
    Case Study: Heartbleed . . . 162

10

UNTRUSTED INPUT 167

Input Validation . . . 168
    Determining Validity . . . 170
    Validation Criteria . . . 170
    Rejecting Invalid Input . . . 171
    Correcting Invalid Input . . . 172

Character String Vulnerabilities 173
Length Issues. 173
Unicode Issues 174
Injection Vulnerabilities 175
SQL Injection 176
Path Traversal 179
Regular Expressions. 181
Dangers of XML 182
Mitigating Injection Attacks 182

11
WEB SECURITY 185

Build on a Framework. 186
The Web Security Model. 187
The HTTP Protocol 188
Digital Certificates and HTTPS 190
The Same Origin Policy 193
Web Cookies 194
Common Web Vulnerabilities. 196
Cross-Site Scripting 196
Cross-Site Request Forgery 199
More Vulnerabilities and Mitigations 201

12
SECURITY TESTING 205

What Is Security Testing?. 206
Security Testing the GotoFail Vulnerability 207
Functional Testing 209
Functional Testing with the Vulnerability 209
Security Test Cases 209
The Limits of Security Tests 210
Writing Security Test Cases 211
Testing Input Validation 211
Testing for XSS Vulnerabilities 212
Fuzz Testing. 214
Security Regression Tests 215
Availability Testing 217
Resource Consumption. 217
Threshold Testing. 218
Distributed Denial-of-Service Attacks 219
Best Practices for Security Testing 219
Test-Driven Development. 219
Leveraging Integration Testing. 220
Security Testing Catch-Up 220

13
SECURE DEVELOPMENT BEST PRACTICES 221

Code Quality. 222
Code Hygiene 222
Exception and Error Handling. 223

    Documenting Security . . . 224
    Security Code Reviews. . . . 224
Dependencies 225
    Choosing Secure Components 225
    Securing Interfaces 226
    Don't Reinvent Security Wheels. . . . 227
    Contending with Legacy Security. . . . 227
Vulnerability Triage 228
    DREAD Assessments 229
    Crafting Working Exploits 230
    Making Triage Decisions 231
Maintaining a Secure Development Environment 231
    Separating Development from Production. . . . 231
    Securing Development Tools 232
    Releasing the Product 232

AFTERWORD 233

Call to Action 234
    Security Is Everyone's Job. . . . 234
    Baking In Security 235
Future Security. . . . 237
    Improving Software Quality 237
    Managing Complexity. . . . 237
    From Minimizing to Maximizing Transparency 238
    Improving Software Authenticity, Trust, and Responsibility 239
Delivering the Last Mile. . . . 240
Conclusion 244

A
SAMPLE DESIGN DOCUMENT 245

Title – Private Data Logging Component Design Document 246
Section 1 – Product Description 246
Section 2 – Overview. . . . 247
    2.1 Purpose 247
    2.2 Scope 247
    2.3 Concepts 247
    2.4 Requirements 248
    2.5 Non-Goals 249
    2.6 Outstanding Issues 249
    2.7 Alternative Designs 249
Section 3 – Use Cases 250
Section 4 – System Architecture 250
Section 5 – Data Design 251
Section 6 – API 252
    6.1 Hello Request 253
    6.2 Schema Definition Request 253
    6.3 Event Log Request 253
    6.4 Goodbye Request 254
Section 7 – User Interface Design. . . . 254

Section 8 – Technical Design 255
Section 9 – Configuration 256
Section 10 – References 256

B
GLOSSARY **257**

C
EXERCISES **269**

D
CHEAT SHEETS **275**

INDEX **281**