

## Auf einen Blick

|    |  |     |
|----|--|-----|
| 1  | Einleitung .....   | 17  |
| 2  | Exploit! So schnell führt ein Programmierfehler zum Root-Zugriff ..... | 21  |
| 3  | Einführung in die sichere Softwareentwicklung .....                    | 43  |
| 4  | Grundlagenwissen für sicheres Programmieren .....                      | 67  |
| 5  | Reverse Engineering .....  | 97  |
| 6  | Sichere Implementierung .....  | 151 |
| 7  | Sicheres Design .....  | 227 |
| 8  | Kryptografie .....   | 281 |
| 9  | Sicherheitslücken finden und analysieren .....                         | 329 |
| 10 | Buffer Overflows ausnutzen .....                                       | 357 |
| 11 | Schutzmaßnahmen einsetzen .....  | 391 |
| 12 | Schutzmaßnahmen umgehen .....  | 401 |
| 13 | Format String Exploits .....   | 451 |
| 14 | Real Life Exploitation .....   | 485 |

# Inhalt

|   |           |
|---|-----------|
| Materialien zum Buch .....  | 13        |
| Geleitwort .....  | 15        |
| <br>  |           |
| <b>1 Einleitung</b> .....   | <b>17</b> |
| <hr/>   |           |
| 1.1 Über dieses Buch .....  | 17        |
| 1.2 Zielgruppe .....  | 19        |
| 1.3 Wie Sie mit dem Buch arbeiten .....   | 20        |
| 1.4 Die Autoren .....   | 20        |
| <br>  |           |
| <b>2 Exploit! So schnell führt ein Programmierfehler zum Root-Zugriff</b> ..... | <b>21</b> |
| <hr/>   |           |
| 2.1 Das Szenario .....  | 21        |
| 2.2 Die Vorbereitungsarbeiten, Informationssammlung .....                       | 22        |
| 2.3 Analyse und Identifikation von Schwachstellen .....                         | 23        |
| 2.4 Ausnutzung der XSS-Schwachstelle .....                                      | 25        |
| 2.5 Analyse und Identifikation weiterer Schwachstellen .....                    | 26        |
| 2.6 Zugriff auf das interne Netzwerk .....                                      | 31        |
| 2.7 Angriff auf das interne Netz .....  | 35        |
| 2.8 Privilege Escalation am Entwicklungsserver .....                            | 39        |
| 2.9 Analyse des Angriffs .....  | 42        |
| <br>  |           |
| <b>3 Einführung in die sichere Softwareentwicklung</b> .....                    | <b>43</b> |
| <hr/>   |           |
| 3.1 Ein Prozessmodell für sichere Softwareentwicklung .....                     | 44        |
| 3.2 Die Praktiken der sicheren Softwareentwicklung .....                        | 46        |

|            |   |           |
|------------|---|-----------|
| 3.2.1      | Code-Review .....   | 47        |
| 3.2.2      | Architectural Risk Analysis .....                             | 47        |
| 3.2.3      | Risk-Based Security-Tests .....                               | 48        |
| 3.2.4      | Penetration Testing .....                                     | 49        |
| 3.2.5      | Abuse Cases .....   | 50        |
| 3.2.6      | Security Operations .....                                     | 51        |
| <b>3.3</b> | <b>Fachwissen für sichere Softwareentwicklung .....</b>       | <b>51</b> |
| 3.3.1      | Injection .....   | 52        |
| 3.3.2      | Broken Authentication .....                                   | 53        |
| 3.3.3      | Sensitive Data Exposure .....                                 | 55        |
| 3.3.4      | XML External Entities (XXE) .....                             | 56        |
| 3.3.5      | Broken Access Control .....                                   | 57        |
| 3.3.6      | Security Misconfiguration .....                               | 59        |
| 3.3.7      | Cross Site Scripting (XSS) .....                              | 60        |
| 3.3.8      | Insecure Deserialization .....                                | 62        |
| 3.3.9      | Using Components with Known Vulnerabilities .....             | 64        |
| 3.3.10     | Insufficient Logging & Monitoring .....                       | 65        |
| <b>4</b>   | <b>Grundlagenwissen für sicheres Programmieren .....</b>      | <b>67</b> |
| <b>4.1</b> | <b>Praktiken der agilen Softwareentwicklung .....</b>         | <b>68</b> |
| <b>4.2</b> | <b>Die Programmiersprache C .....</b>                         | <b>69</b> |
| 4.2.1      | Ein einfaches C-Programm .....                                | 70        |
| 4.2.2      | Automatisierung des Build-Prozesses .....                     | 72        |
| 4.2.3      | Die Erstellung und Verwendung von Bibliotheken in C .....     | 73        |
| <b>4.3</b> | <b>Die Programmiersprache Java .....</b>                      | <b>76</b> |
| 4.3.1      | Ein einfaches Java-Programm .....                             | 76        |
| 4.3.2      | Automatisierung des Build-Prozesses .....                     | 77        |
| 4.3.3      | Die Erstellung und Verwendung von Bibliotheken in Java .....  | 79        |
| <b>4.4</b> | <b>Versionierung von Quellcode .....</b>                      | <b>82</b> |
| <b>4.5</b> | <b>Debugging und automatisiertes Testen .....</b>             | <b>84</b> |
| 4.5.1      | Debugging .....   | 85        |
| 4.5.2      | Automatisiertes Testen .....                                  | 85        |
| <b>4.6</b> | <b>Continuous Integration .....</b>                           | <b>91</b> |
| <b>4.7</b> | <b>Beispiele auf GitHub und auf rheinwerk-verlag.de .....</b> | <b>94</b> |

|            |   |            |
|------------|---|------------|
| <b>5</b>   | <b>Reverse Engineering .....</b>                                | <b>97</b>  |
| <b>5.1</b> | <b>Analyse von C-Applikationen .....</b>                        | <b>97</b>  |
| 5.1.1      | Die x86-Architektur .....                                       | 97         |
| 5.1.2      | Die x86-64-Architektur .....                                    | 100        |
| 5.1.3      | Speicheraufteilung von C-Applikationen .....                    | 101        |
| 5.1.4      | Der GNU Debugger .....  | 102        |
| 5.1.5      | Die Verwendung des Heap .....                                   | 106        |
| 5.1.6      | Die Verwendung des Stacks .....                                 | 112        |
| 5.1.7      | Reverse Engineering: Ein konkretes Beispiel .....               | 120        |
| <b>5.2</b> | <b>Analyse von Java-Applikationen .....</b>                     | <b>129</b> |
| 5.2.1      | Classdateiformat .....  | 130        |
| 5.2.2      | Speicheraufteilung von Java-Applikationen .....                 | 132        |
| 5.2.3      | Befehlssatz .....   | 133        |
| 5.2.4      | Der Java-Disassembler .....                                     | 135        |
| 5.2.5      | Class Loader .....  | 136        |
| 5.2.6      | Garbage Collectors .....  | 139        |
| 5.2.7      | Just-in-Time Compiler .....                                     | 141        |
| 5.2.8      | Reverse Engineering Java: Ein konkretes Beispiel .....          | 143        |
| <b>5.3</b> | <b>Code Obfuscation .....</b>                                   | <b>148</b> |
| <b>6</b>   | <b>Sichere Implementierung .....</b>                            | <b>151</b> |
| <b>6.1</b> | <b>Reduzieren Sie die Sichtbarkeit von Daten und Code .....</b> | <b>151</b> |
| 6.1.1      | Zugriffskontrollen in Java .....                                | 152        |
| 6.1.2      | Objekte als Parameter .....                                     | 153        |
| 6.1.3      | Object Serialization .....                                      | 157        |
| 6.1.4      | Immutable Objects .....   | 158        |
| 6.1.5      | Java Reflection API .....                                       | 159        |
| <b>6.2</b> | <b>Der sichere Umgang mit Daten .....</b>                       | <b>160</b> |
| 6.2.1      | Repräsentation von Daten .....                                  | 161        |
| 6.2.2      | Input-Validierung .....   | 170        |
| 6.2.3      | Output-Codierung .....  | 174        |
| <b>6.3</b> | <b>Der richtige Umgang mit Fehlern .....</b>                    | <b>176</b> |
| 6.3.1      | Fehlercodes .....   | 176        |
| 6.3.2      | Exceptions .....  | 177        |
| 6.3.3      | Logging .....   | 179        |

|            |  |     |
|------------|--|-----|
| <b>6.4</b> | <b>Kryptografische APIs richtig einsetzen</b> .....              | 182 |
| 6.4.1      | Java Cryptography Architecture .....                             | 183 |
| 6.4.2      | Sichere Datenspeicherung .....                                   | 185 |
| <b>6.5</b> | <b>Statische Codeanalyse</b> .....                               | 211 |
| 6.5.1      | Manuelles Code-Review .....                                      | 212 |
| 6.5.2      | Automatisiertes Code-Review .....                                | 213 |
| 6.5.3      | Analyse von Bibliotheken .....                                   | 223 |
| <b>7</b>   | <b>Sicheres Design</b> .....                                     | 227 |
| <b>7.1</b> | <b>Architekturbasierte Risikoanalyse</b> .....                   | 227 |
| 7.1.1      | Analyse der Angriffsfläche .....                                 | 228 |
| 7.1.2      | Bedrohungsmodellierung .....                                     | 229 |
| <b>7.2</b> | <b>Designprinzipien für sichere Softwaresysteme</b> .....        | 232 |
| <b>7.3</b> | <b>Das HTTP-Protokoll</b> .....                                  | 235 |
| 7.3.1      | HTTP-Transaktionen .....   | 236 |
| 7.3.2      | Cookies .....  | 239 |
| 7.3.3      | HTTPS .....  | 240 |
| 7.3.4      | Interception Proxy .....   | 242 |
| <b>7.4</b> | <b>Sicheres Design von Webapplikationen</b> .....                | 244 |
| 7.4.1      | Clientseitige Kontrolle .....                                    | 244 |
| 7.4.2      | Zugriffskontrolle .....  | 248 |
| 7.4.3      | Authentifizierung .....  | 249 |
| 7.4.4      | Session-Management .....   | 254 |
| 7.4.5      | Autorisierung .....  | 257 |
| 7.4.6      | Datenspeicherung .....   | 263 |
| 7.4.7      | Verhinderung von Browserangriffen .....                          | 271 |
| <b>8</b>   | <b>Kryptografie</b> .....  | 281 |
| <b>8.1</b> | <b>Verschlüsselung</b> .....                                     | 281 |
| 8.1.1      | Grundbegriffe .....  | 281 |
| 8.1.2      | Symmetrische Verschlüsselung: Designideen von AES .....          | 290 |
| 8.1.3      | Asymmetrische Verschlüsselung: Hinter den Kulissen von RSA ..... | 298 |

|            |   |     |
|------------|---|-----|
| 8.1.4      | RSA: Security-Überlegungen .....                                  | 304 |
| 8.1.5      | Diffie-Hellman Key Exchange .....                                 | 305 |
| <b>8.2</b> | <b>Hash-Funktionen</b> .....                                      | 309 |
| 8.2.1      | Grundlegende Eigenschaften von Hash-Funktionen .....              | 309 |
| 8.2.2      | Angriffe auf Hash-Funktionen .....                                | 313 |
| 8.2.3      | Ausgewählte Architekturen von Hash-Funktionen .....               | 319 |
| <b>8.3</b> | <b>Message Authentication Codes und digitale Signaturen</b> ..... | 321 |
| 8.3.1      | Message Authentication Codes .....                                | 322 |
| 8.3.2      | Digitale Signaturen .....   | 324 |
| <b>8.4</b> | <b>NIST-Empfehlungen</b> .....                                    | 327 |
| <b>9</b>   | <b>Sicherheitslücken finden und analysieren</b> .....             | 329 |
| <b>9.1</b> | <b>Installation der Windows-Testinfrastruktur</b> .....           | 329 |
| 9.1.1      | Installation des i.Ftp-Clients .....                              | 330 |
| 9.1.2      | Installation der Debugging-Umgebung .....                         | 332 |
| 9.1.3      | Installation der PyWin-Umgebung .....                             | 334 |
| <b>9.2</b> | <b>Manuelle Analyse der Anwendung</b> .....                       | 335 |
| 9.2.1      | Identifikation von Datenkanälen in die Anwendung .....            | 336 |
| 9.2.2      | Analyse der XML-Konfigurationsdateien .....                       | 339 |
| <b>9.3</b> | <b>Automatische Schwachstellensuche mittels Fuzzing</b> .....     | 340 |
| <b>9.4</b> | <b>Analyse des Absturzes im Debugger</b> .....                    | 343 |
| <b>9.5</b> | <b>Alternativen zum Fuzzing</b> .....                             | 344 |
| <b>9.6</b> | <b>Tools zur Programmanalyse</b> .....                            | 344 |
| 9.6.1      | Olly Debug .....  | 345 |
| 9.6.2      | Immunity Debugger .....   | 348 |
| 9.6.3      | WinDebug .....  | 348 |
| 9.6.4      | x64dbg .....  | 349 |
| 9.6.5      | IDA Pro .....   | 351 |
| 9.6.6      | Hopper .....  | 351 |
| 9.6.7      | GDB – Gnu Debugger .....  | 353 |
| 9.6.8      | EDB – Evan’s Debugger .....                                       | 353 |
| 9.6.9      | Radare2 .....   | 354 |
| 9.6.10     | Zusammenfassung der Tools .....                                   | 355 |

|              |  |     |
|--------------|--|-----|
| <b>10</b>    | <b>Buffer Overflows ausnutzen</b>              | 357 |
| <b>10.1</b>  | <b>Die Crash-Analyse des i.Ftp-Clients</b>     | 357 |
| <b>10.2</b>  | <b>Offsets ermitteln</b>                       | 360 |
| <b>10.3</b>  | <b>Eigenen Code ausführen</b>                  | 363 |
| <b>10.4</b>  | <b>Umgang mit Bad Characters</b>               | 368 |
| <b>10.5</b>  | <b>Shellcode generieren</b>                    | 372 |
| 10.5.1       | Bind Shellcode                                 | 372 |
| 10.5.2       | Reverse Shellcode                              | 375 |
| <b>10.6</b>  | <b>Exception Handling ausnutzen</b>            | 377 |
| <b>10.7</b>  | <b>Analyse unterschiedlicher Buffer-Längen</b> | 379 |
| <b>10.8</b>  | <b>Buffer Offsets berechnen</b>                | 382 |
| <b>10.9</b>  | <b>SEH-Exploits</b>                            | 382 |
| <b>10.10</b> | <b>Heap Spraying</b>                           | 386 |
| <b>11</b>    | <b>Schutzmaßnahmen einsetzen</b>               | 391 |
| <b>11.1</b>  | <b>ASLR</b>                                    | 391 |
| <b>11.2</b>  | <b>Stack Cookies</b>                           | 393 |
| <b>11.3</b>  | <b>SafeSEH</b>                                 | 395 |
| <b>11.4</b>  | <b>SEHOP</b>                                   | 396 |
| <b>11.5</b>  | <b>Data Execution Prevention</b>               | 396 |
| <b>11.6</b>  | <b>Schutz gegen Heap Spraying</b>              | 399 |
| <b>12</b>    | <b>Schutzmaßnahmen umgehen</b>                 | 401 |
| <b>12.1</b>  | <b>Was sind Reliable Exploits?</b>             | 401 |
| <b>12.2</b>  | <b>Bypass von ASLR</b>                         | 402 |
| 12.2.1       | Review des i.Ftp-Exploits                      | 402 |
| 12.2.2       | Verwendung von ASLR-freien Modulen             | 404 |
| 12.2.3       | Verwendung von ASLR-freien Modulen – i.Ftp.exe | 405 |
| 12.2.4       | Partielles Überschreiben der Sprungadresse     | 408 |

|              |   |     |
|--------------|---|-----|
| 12.2.5       | Egg Hunting   | 411 |
| 12.2.6       | Verwendung von ASLR-freien Modulen – Lgi.dll  | 416 |
| 12.2.7       | Brute Force einer Sprungadresse   | 417 |
| 12.2.8       | Partielles Überschreiben der Return-Adresse II                                      | 417 |
| 12.2.9       | Ermitteln der Adressen aus dem laufenden Programm                                   | 417 |
| 12.2.10      | Fehlertolerante Sprungbereiche  | 418 |
| <b>12.3</b>  | <b>Bypass von Stack Cookies</b>   | 418 |
| <b>12.4</b>  | <b>Bypass von SafeSEH</b>   | 419 |
| <b>12.5</b>  | <b>Bypass von SEHOP</b>   | 420 |
| <b>12.6</b>  | <b>Data Execution Prevention (DEP) – Bypass mittels Return Oriented Programming</b> | 423 |
| 12.6.1       | DEP unter Windows   | 424 |
| 12.6.2       | Return Oriented Programming   | 429 |
| <b>12.7</b>  | <b>DEP Bypass mittels Return-to-libc</b>  | 449 |
| <b>13</b>    | <b>Format String Exploits</b>   | 451 |
| <b>13.1</b>  | <b>Formatstrings</b>  | 451 |
| <b>13.2</b>  | <b>Die fehlerhafte Anwendung</b>  | 454 |
| <b>13.3</b>  | <b>Aufbau der Analyseumgebung</b>   | 457 |
| <b>13.4</b>  | <b>Analyse des Stack-Inhalts</b>  | 459 |
| <b>13.5</b>  | <b>Speicherstellen mit %n überschreiben</b>   | 463 |
| <b>13.6</b>  | <b>Die Exploit-Struktur</b>   | 466 |
| <b>13.7</b>  | <b>Die Ermittlung von Adressen und Offsets</b>                                      | 468 |
| <b>13.8</b>  | <b>Die Verifikation der Adressen im Debugger</b>                                    | 471 |
| <b>13.9</b>  | <b>Die erste lauffähige Version des Exploits</b>                                    | 473 |
| <b>13.10</b> | <b>ASLR Bypass</b>  | 477 |
| <b>14</b>    | <b>Real Life Exploitation</b>   | 485 |
| <b>14.1</b>  | <b>Heartbleed</b>   | 485 |
| <b>14.2</b>  | <b>SSL OpenFuck</b>   | 488 |

|  |     |
|--|-----|
| <b>14.3 Shellshock</b> .....           | 493 |
| <b>14.4 Eternal Blue</b> .....         | 495 |
| <b>14.5 Spectre und Meltdown</b> ..... | 504 |
| <b>14.6 Stagefright</b> .....          | 509 |
| <br>                                   |     |
| Index .....                            | 511 |