

INDEX

A

- ABAC (attribute-based access control), 17
- Accept Security Responsibility pattern, 69
- access control list (ACL), 55
- access controls, 48
- access policy
 - custom, 48
 - fine-grained, 49
 - “relief valve”, 49
- Advanced Encryption Standard (AES), 82
- adversaries, 24
 - attacker’s advantage, 25
 - targets, 25
- Agarkov, Dmitry, 175
- allowlists, 60. *See also* blocklists
- Allowlists over Blocklists pattern, 60
- anti-patterns
 - Backflow of Trust, 73
 - Confused Deputy, 71, 224
 - Intention and Malice, 72
 - Trustworthy Deputy, 73
 - Security by Obscurity, 135
 - Third-Party Hooks, 74
 - Unpatchable Components, 74
- antivirus, 61, 225
- “Are you sure?” dialog, 70
- Ariane 5, 223
- arithmetic
 - 32-bit, 149
 - 64-bit, 154
 - binary, 148
 - floating-point vs. integer, 146
 - modular, 147
 - safe, 155
 - vulnerabilities, 146–156
- assessment report. *See* security design review (SDR)
- assets, 26
 - aggregation, 29
 - differing valuation, 29
 - identification, 28
 - isolation of, 120
 - removal, 38
 - valuation, 25
- atomicity, 140
- attacks
 - denial-of-service (DoS), 13, 216
 - distributed denial-of-service (DDoS), 219
 - injection, 175
 - preimage, 79
 - replay, 79, 82
 - side-channel, 11, 30, 141
 - timing, 141–142
- attack surfaces, 26, 101, 113, 119
 - hardening, 45
 - identification, 30
 - internet, 30
 - minimization, 45
- attribute-based access control (ABAC), 17
- auditing, 14
 - inside jobs, 18
 - shared account problem, 19
- audit logs, 14
 - Goldilocks principle, 19
 - need for monitoring, 19
 - non-repudiability, 19
 - private information problem, 19
 - tamper-resistant, 18
- authentication, 14
 - binding attacks, 17
 - binding the principal, 16

- authentication (*continued*)
 - separation from authorization, 15
 - something you are, 15
 - something you have, 15
 - something you know, 15
 - somewhere you are, 15
 - authN. *See* authentication
 - authorization, 14
 - anonymous, 17
 - attribute-based access control (ABAC), 17
 - guards, 17
 - minimal access, 18
 - multiple principals, 18
 - no self-service, 18
 - policy-based access control (PBAC), 17
 - rate-limited, 18
 - role-based access control (RBAC), 17
 - time of day, 18
 - authZ. *See* authorization
 - availability, 11, 13
 - availability testing, 216–218
 - Avoid Predictability pattern, 61
- B**
- Backflow of Trust anti-pattern, 73
 - backups, 13
 - BeautifulSoup parser, 214
 - binary math refresher, 148
 - Bloch, Josh, 97
 - blocklists, 60. *See also* allowlists
 - bottleneck. 63
 - bridge (between user processes), 64
 - brute-force guessing, 16
 - buffer overflow, 157
 - example, 158
 - buffer overrun, 157, 158
- C**
- California Senate Bill No. 327 (2018), 59
 - CAS. *See* Code Access Security (CAS)
 - case study
 - difficult SDR, 122
 - GotoFail vulnerability, 137
 - Heartbleed vulnerability, 162
 - The Most Dangerous Code in the World, 226
 - certificate authority (CA), 87
 - chosen plaintext attack, 84
 - C-I-A, 11–14, 99
 - principles. *See also* information security
 - ciphertext, 81, 84
 - Code Access Security (CAS), 73, 243
 - .NET Framework, 46
 - permission, 46
 - code examples, 130
 - code quality
 - code hygiene, 222
 - documentation, 224
 - exception handling, 223
 - security reviews, 224
 - collision attack, 78
 - competence and imperfection, 7
 - compiler warnings, 139, 222
 - Complete Mediation pattern, 63
 - degrees of compliance, 63
 - components
 - security considerations, 225
 - selecting, 225
 - confidentiality, 11–13
 - compromise of, 12
 - Confused Deputy anti-pattern, 71, 224
 - Intention and Malice, 72
 - Trustworthy Deputy, 73
 - cookies. *See* HTTP protocol
 - C programming language, 130, 138
 - credentials, 14. *See also* authentication
 - Cross-Origin Resource Sharing (CORS), 196
 - cross-site request forgery (CSRF or XSRF), 201. *See also* web security
 - example, 199
 - mitigation, 200
 - cross-site scripting (XSS). *See also* web security
 - DOM-based, 199
 - example, 197

- mitigation, 199, 202
- reflected, 198
- stored, 198
- testing, 212–214
- cryptocurrency, 86
- cryptographically secure pseudo-
 - random number generators (CSPRNG), 77
- cryptography. *See* encryption
- crypto toolbox, 76, 89
- CSRF. *See* cross-site request forgery (CSRF or XSRF)
- Cuban, Mark, 121

D

- data
 - backups, 13
 - invisibility of, 6
 - private, 12
 - provenance, 13
 - tampering, 13
- data flow diagrams (DFD), 27
- data hiding, 73
- data protection
 - backups, 51
 - data at rest, 51
 - minimizing data exposure, 47
 - offline backups, 48
- DDoS (distributed denial-of-service)
 - attacks, 219
- deanonymization, 12–13
- default password, 59
- Defense in Depth pattern, 65
- denial of service (DoS). *See* availability testing, STRIDE
- denial-of-service (DoS) attacks, 13, 181, 206
- dependencies, 228
 - choosing components, 225
 - legacy code, 227
 - libraries and frameworks, use of, 227
 - secure design, 99
 - secure interfaces, 226
 - software supply chain, 225
- deprecation, 226
- DES encryption algorithm, 56
- deserialization, 143

- design. *See also* secure design
 - common assumptions, 98
 - documents, 97
 - importance of assumptions, 98
 - integrating security, 96
 - scope, 98
 - consequences of not defining, 99
 - looking beyond, 99
 - security considerations, 97
- design pattern groupings, 54
- DFD (data flow diagrams), 27
- dialog fatigue, 70
- Diffie–Hellman key exchange
 - algorithm, 88
- Diffie, Whitfield, 87
- digest for integrity, 13
- digital certificate. *See* HTTPS protocol
- digital signature
 - algorithm, 85
 - for integrity, 13
 - signature verification, 85
- distributed denial-of-service (DDoS)
 - attacks, 219
- documentation for security, 224
- Document Object Model (DOM), 193
- Domain Name System (DNS), 189
- DoS (denial-of-service), 13
- downgrade attack, 192
- DREAD model
 - example, 229
 - T-shirt sizes, 229
- dynamic memory allocation, 157

E

- Economy of Design pattern, 54, 108
- electronic code book (ECB) mode, 82
- elevation of privilege. *See* STRIDE
- elliptic curve algorithms, 85
- email retention, 21
- encryption
 - asymmetric
 - elliptic curve, 85
 - private key, 83
 - public key, 83
 - RSA, 84
 - backup data application, 90
 - ciphertext, 81

- encryption (*continued*)
 - communication, 50
 - cryptocurrency application, 90
 - digital signatures, 85
 - ECB mode, 82
 - exclusive-or, 81
 - financial data application, 90
 - foundations, 91
 - limitations of, 91
 - plaintext, 81
 - symmetric, 81
 - AES, 82
 - block cipher, 82
 - key establishment, 83
 - key secrecy, 83
 - key size, 83
 - limitations of, 83
 - entropy, 136
 - sources, 78
 - Equifax breach, 107
 - error handling, 223
 - eval function, 184
 - examples
 - accountant, 64
 - Ariane 5, 223
 - backing up photos, 51
 - bank vault, 33–35
 - child-proofing, 40
 - COVID-19 stay-at-home emergency order, 60
 - credit card contract, 175
 - customer relationship management (CRM), 58
 - endianness mix-up, 97
 - floating-point underflow, 151
 - generating random numbers using lava lamps, 78
 - HTTP cookies, 68
 - iMessage, 64
 - integer overflow, 153
 - LEGO, 55
 - memory allocation vulnerabilities, 158
 - Norman Bates, 71
 - Ocean’s Eleven, 37
 - online shopping app with bugs, 134
 - plywood, 65
 - predictable account IDs, 61
 - Reddit user, 64
 - safe deposit box, 67
 - Star Wars, 56
 - Superman, 57
 - tax ID privacy, 63
 - traveling sales staff, 101
 - exception handling, 223
- F**
- Facebook Beacon, 30
 - Fail Securely pattern, 62
 - floating point
 - equality test problematic, 150
 - precision, 149
 - Python example, 151
 - footguns, 138-139
 - Four Questions, 25, 98–99, 103
 - as guidance for a security design review, 116–119
 - free function, 157
 - functional testing, 209
 - with GotoFail vulnerability, 209
 - fuzz testing, 215
 - example, 214
- G**
- Garg, Praerit, 35
 - GCC compiler, 139
 - General Data Protection Regulation (GDPR), 12
 - Goldilocks principle, 28
 - Gold Standard, 11, 16–19, 37
 - auditing, 14
 - authentication, 14
 - authorization, 14
 - meaning of name, 14
 - relation to C-I-A, 15
 - GotoFail vulnerability, 137, 140
 - lessons, 139
 - source code, 138
 - guard, 63
- H**
- hardware random number generators (HRNG), 78

- hash. *See* message digest
 - SHA-256, 200
- heap, 157
- heartbeat, TLS, 162
- Heartbleed vulnerability, 47
- Hellman, Martin, 87
- homomorphs, 174
- HTTP over TLS/SSL. *See* HTTPS
 - protocol
- HTTP protocol, 188
 - cookie attributes
 - httponly, 195
 - SameSite, 201
 - secure, 195
 - cookies
 - session, 194–195, 198, 200
 - sharing, 195
 - Cross-Origin Resource Sharing (CORS), 196
 - GET, 189, 199
 - POST, 189, 199–200
 - request headers, 189
 - REFERER, 189, 202
 - response headers, 189
 - Content-Security-Policy, 202
 - Referrer-Policy, 190, 202
 - security-related, 202
 - verbs, 189
- HTTPS protocol
 - adoption of, 190
 - cipher suites, 193
 - digital certificates
 - Let’s Encrypt, 192
 - types of, 192
 - downgrade attacks, 192
 - security properties, 191
 - Strict-Transport-Security
 - directive, 193

I

- identity management, 16
- IEEE 754. *See* floating point
- IMDb, 13
- implementation from design, 129
- influencing code, 131
- information collection, 20
- information disclosure. *See* STRIDE

- information security, 5
 - principles (C-I-A), 11–14
 - relation to authorization, 11
- injection attacks, 175
 - avoiding, 183
 - backtracking regex, 181, 217
 - cross-site scripting, 196–199
 - mitigation, 183
 - “No Game Scheduled”, 176
 - path traversal, 179
 - shell command, 183
 - SQL, 176–179
 - XML entities, 182
- input validation, 168
 - character string length, 173
 - correcting invalid input, 172
 - range check, 169
 - rejecting invalid input, 171
 - requirements, 170
 - size check, 170
 - Unicode issues, 174
 - valid for a purpose, 171
- inside jobs, 18
- insurance, 38
- integer overflow, 146
 - mitigation, 155
 - security testing, 206
- integration testing
 - data leak detection, 220
- integrity, 11, 13
- Intention and Malice. *See* Confused
 - Deputy anti-pattern
- interfaces
 - between components, 225
 - intraprocess, 50
 - kinds of, 49
 - secure design, 103
 - securing, 226
- Internet Explorer, 35

K

- keyed hash function, 79
- key exchange, 87
 - Diffie–Hellman algorithm, 88
 - randomness requirement, 89
 - secure communication
 - establishment, 89

L

- last mile, 240
- leaks, memory, 160
- Least Common Mechanism pattern, 64, 108
- Least Information pattern, 57, 104
- Least Privilege pattern, 56, 178
- legacy security, 227
- Let's Encrypt, 192
- loopholes, 62
- low-level programming, 146

M

- malloc function, 157, 160
- managing complexity, 237
- math.isclose function, 150
- Meltdown, 141
- memory
 - access vulnerabilities, 156–162
 - buffer overflow, 157
 - heap, 157
 - leaks, 160
 - management, 156
- message authentication code (MAC), 78
 - nonce, 80
 - replay attacks, 79
 - secure communications use, 80
 - tamper prevention, 79
- message digest, 78–80
 - collision, 78
 - replay attacks, 79
- Microsoft Windows, 35
- Minsky, Marvin, 136
- misleading indentation warning, 139
- mitigation, 38, 43–52
 - definition of, 44
 - minimizing attack surfaces, 45
 - minimizing data exposure, 47
 - narrowing windows of vulnerability, 46
 - partial, 39
 - protecting communications, 50
 - protecting interfaces, 49
 - protecting storage, 51
 - real-world examples of, 43
 - structural, 45–48

- mobile data security, 241
- models
 - Code Access Security (CAS), 46, 73
 - data flow diagrams (DFD), 27
 - Unified Modeling Language (UML), 27
- Morris worm, 233

N

- National Security Agency (NSA), 100
- National Transportation Safety Board (NTSB), 239
- Netflix, 13
- .NET Framework, 46, 243
- Netscape Navigator, 35
- nonce, 80, 201

O

- obsolescence
 - software support, 52
 - storage media, 51
- one-time pad, 81
 - reuse problem, 82
 - use restrictions, 82
- OpenSSL, 162
- opportunistic protection, 29
- overflow
 - buffer, 157
 - integer, 146
 - common vulnerabilities, 149
 - example, 153
 - mitigation, 155

P

- padding, 80
- path traversal, 179
- patterns
 - Accept Security Responsibility, 69
 - Allowlists over Blocklists, 60
 - Avoid Predictability, 61
 - Complete Mediation
 - degrees of compliance, 63
 - Defense in Depth, 65
 - design attributes, 54–56
 - Economy of Design, 54, 108
 - exposure minimization, 56–62
 - Fail Securely, 62

- general use of, 54
- Least Common Mechanism, 64, 108
- Least Information, 57, 104
- Least Privilege, 56
- redundancy, 65–68
- Reluctance to Trust, 68
- Secure by Default, 59, 226
- Separation of Duty, 232
- Separation of Privilege, 67
- strong enforcement, 62–65
- Transparent Design, 56, 77
- trust and responsibility, 68–70
- personal data
 - collection, 39
 - disclosure mitigation, 40
- personally identifiable information (PII), 102
- plaintext, 81, 84
- policy-based access control (PBAC), 17
- preimage attack, 79
- principal, 14
- principles of information security
 - availability, 11, 13
 - confidentiality, 11
 - integrity, 11, 13
- privacy, 39
 - email retention, 21
 - human factors, 20
 - information collection, 20
 - policy, 21
 - relation to security, 19
 - software security challenges, 20
- privacy policy, 120
 - auditing, 105
 - explicit protection, 105
 - owner, 105
- privacy reviews, 120
- private data, 12
- private key, 83
- provenance, 13, 240
- pseudo-random number generators (PRNG), 77
- pseudo-random numbers, 77. *See also* random numbers
 - cryptographically secure, 77

- public key, 83
- Pwn2Own competitive hacking contest, 135
- Python programming language, 130
 - structuring by indentation, 138

R

- random numbers
 - applications, 77
 - classes, 77
 - cryptographically secure pseudo-random number generators, 77
 - entropy sources, 78
 - hardware random number generators, 78
 - pseudo-random number generators, 77
 - unpredictability, 77
- RBAC (role-based access control), 17
- regular expressions (regex)
 - backtracking, 181, 217
- reidentification, 13
- Reluctance to Trust pattern, 68
- replay attacks, 79, 82
- repudiation, 37. *See also* STRIDE
- risk acceptance, 38
- risk assessment, 29
 - T-shirt sizes, 29, 229–230
- risk transfer, 38
- role-based access control (RBAC), 17
- root certificate, 87
- RSA cryptosystem
 - algorithm, 84
 - history, 84
 - mathematical basis, 84

S

- Same Origin Policy (SOP), 193–196
 - CSRF vulnerability, 199
- sample design document, 19, 96, 245
- sandbox, 65
- SDR. *See* security design review (SDR)
- Secure by Default pattern, 59, 226
- secure design, 95–108
 - balanced approach, 102
 - cache implications, 102
 - data handling, 104

- secure design (*continued*)
 - dependencies, 99
 - design assumptions, 97
 - examples, 98
 - importance of making
 - explicit, 97
 - end of life, 106
 - exploring alternatives, 107
 - high security requirements, 100
 - interfaces, 103
 - minimal security requirements, 100
 - mitigation, 103
 - privacy, 105
 - requirements statements, 100
 - sample design document, 19,
 - 96, 245
 - scope definition
 - importance, 98
 - iterative design, 99
 - software lifecycle, 106
 - trade-offs, 106
- secure development environment, 231
- securely random IDs, 62
- secure programming, 130
- security
 - goals, 36
 - information, 5
 - mindset, 23
 - physical, 4
 - software, 5
 - trust but verify, 8
 - understanding, 4
- Security by Obscurity anti-pattern,
 - 56, 135
- security code reviews, 224
- security design review (SDR), 109–125
 - assessment report, 114
 - minimal, 115
 - organization, 115
 - Recommendations Declined
 - section, 123
 - benefits of, 110
 - collaboration with designer, 113
 - design updates, 120
 - documentation, 111
 - guidance, 116–119
 - importance of context, 117
 - incremental updates, 120
 - independent reviewer, 109
 - logistics, 110
 - managing disagreements, 121–124
 - escalation, 123
 - meeting preparation, 123
 - missing mitigations, 118
 - practicing, 124
 - problem solving, 122
 - process, 111
 - progress tracking, 116
 - recommendation ranking, 114
 - relation to secure design, 95
 - reviewer role, 115
 - sandwich method feedback, 122
 - separate from functional
 - review, 110
 - showing versus telling, 123
 - stages, 111–116
 - summary statement, 119
 - tactful communication, 121
 - threat identification, 117
 - timing, 110
 - ways to practice, 124
 - where to dig, 119
- security regression tests
 - Heartbleed example, 216
 - how to write, 216
 - importance, 215
- security requirements
 - data collection, 101
 - high-value private key, 101
 - top-secret document, 100
- security testing, 205–220
 - best practices, 219
 - catching up, 220
 - cross-site scripting, 212
 - denial-of-service attacks, 216
 - exception handling, 206
 - GotoFail vulnerability, 207, 209
 - importance of, 207
 - input validation, 211
 - integer overflow, 206
 - limits of, 210
 - memory management, 206
 - resource consumption, 217
 - threshold testing, 218

- untrusted inputs, 206
- web security, 206
- writing test cases, 211
- Separation of Duty pattern, 67, 232
- Separation of Privilege pattern, 67
- serialization, 143
- SHA-256 hash, 200
- Shostack, Adam, 25
- side-channel attack, 11, 30, 141
- Snowden, Edward, 100
- software quality, 237
- software security, 5
- software supply chain, 225
- SOP (Same Origin Policy), 193–196
- Spectre, 141
- speculative execution, 141
- spoofing, 36. *See also* STRIDE
- SQL injection, 176–179
- stories
 - auto salesman, 4
 - driver’s ed, 75
 - “No Game Scheduled”, 176
 - street crossing, 6
- strcpy function, 161
- STRIDE, 35–38
 - definition, 35
 - origins, 35
 - relation to information security
 - principles, 37
 - repudiation, 37
- strncpy function, 161
- strtol function, 160
- sudo, 57

T

- tainting, 132
- tampering, 13, 37, 78, 143. *See also* STRIDE
 - prevention with MAC, 79
- Taylor, Jason, 229
- test-driven development (TDD), 219
- The Most Dangerous Code in the World, 226
- Third-Party Hooks anti-pattern, 74. *See also* Backflow of Trust anti-pattern
- Thompson, Ken, 240

- threat modeling, 78, 101–103
 - asset prioritization, 29
 - balancing security needs, 102
 - definition, 26
 - early efforts, 24
 - essential threat model, 102
 - granularity, 28
 - incorporating into design, 101
 - iterative process, 27
 - methodology varieties, 27
 - overview, 26
 - personally identifiable information, 102
 - real-life applications, 41
 - real world, 40
 - real world versus digital, 27
 - working from a model, 27
- threats, 23–41 *See also* attacks
 - addressing, 44
 - availability, 13
 - brute-force guessing, 16
 - categorizing with STRIDE, 35
 - fact of communication, 50
 - identifying, 33
 - mitigation, 38, 43–52
 - privacy, 39
- threat taxonomy. *See* STRIDE
- timing attack
 - forgot password example, 142
 - Meltdown, 141
 - mitigation, 142
 - Spectre, 141
 - speculative execution example, 141
- toolbox. *See* crypto toolbox
- transparency, 238
- Transparent Design pattern, 56, 77
- Transport Layer Security (TLS), 89, 162
 - Heartbeat Extension, 162
- triage. *See* vulnerability triage
- trust, 5
 - actions, 10
 - being trustworthy, 10
 - decisions, 8
 - decision tree, 8
 - features, 10
 - feeling trust, 6

- trust (*continued*)
 - independent third-party, 10
 - spectrum, 8
 - transparency, 10
 - trust but verify, 8
- trust boundaries, 26, 101, 120
 - identification, 30
 - kernel/userland interface, 31
- trust level
 - aggregating or splitting, 32
 - trust vs. privilege, 31
- Trustworthy Deputy. *See also* Confused Deputy anti-pattern
- Twitter, 19

U

- underflow, 150
 - mitigation, 152
- understanding security, 4–5
- Unicode
 - case, changing, 175
 - combining characters, 175
 - homomorphs, 174
- Unified Modeling Language (UML), 27
- uniform resource locator (URL), 188
- Unpatchable Components
 - anti-pattern, 74
- unpickling, 143
- untrusted input, 132, 143, 167–168
- userland. *See* trust boundaries

V

- vulnerabilities, 130, 133.
 - buffer overflow, 160
 - character string, 173–175
 - countermeasures, 140
 - cross-site request forgery (CSRF) or XSRF, 199
 - cross-site scripting, 196
 - example of a chain, 134
 - fixed-width integer, 147
 - floating point, 149
 - GotoFail, 137
 - Heartbleed, 162, 216
 - injection, 175, 199
 - path traversal, 179

- regular expressions, 181
- relation to bugs, 133
- SQL injection, 176–179
- Unicode, 174
- XML entities, 182

- vulnerability, narrowing windows
 - of, 46
- vulnerability chains, 134
- vulnerability triage, 228–231
 - crafting working exploits, 230
 - decision making, 231
 - DREAD assessments, 229

W

- web security, 185–203
 - client/server model, 187
 - common vulnerabilities, 196–201
 - CSS visited selector, 202
 - frameworks, 186
 - HTML5, 196
 - HTTP header injection, 202
 - HTTP response headers, 202
 - model, 187
 - redirects, 202
 - rel="noopener" attribute, 202
 - rel="noreferrer" attribute, 202
 - session cookies, 194–195, 200
 - X-Frame-Options header, 202
 - XML external entity
 - attacks, 202
- window.open, 193
- World Wide Web, 185. *See also* web security

X

- xkcd comics
 - Epoch fail (376), 219
 - Exploits of a Mom (327), 176
 - Heartbleed Explanation (1354), 165
 - Security versus the \$5 wrench (538), 90
- XSRF. *See* cross-site request forgery (CSRF or XSRF)
- XSS. *See* cross-site scripting (XSS)