INDEX

Symbols

`../`, 279, 287, 325
.bash_profile, 81
/etc/passwd, 252, 291
/etc/shadow, 177, 249, 253–260, 279, 332
.git directory, 328–330. *See also* Git
  annotated tags, 330
  blobs, 330
  commits, 330
  trees, 330

A

access control, 43, 175, 177–178, 278, 324, 364–365. *See also* broken access control
access tokens, 312–316, 364–365
  long-lived tokens, 316
account takeover, 172, 185, 321
active scanning, 69. *See also* passive scanning
ADB. *See* Android Debug Bridge (ADB)
admin panels, 70–71, 278, 321
AFL. *See* American Fuzzy Lop (AFL)
alert box, 116, 122–126
allowlist, 133, 141, 194, 215, 220–221. *See also* blocklist
Altdns, 69
Amass, 68
Amazon Elastic Compute Cloud (EC2), 77, 226. *See also* Amazon Web Services (AWS)
Amazon S3, 74–77, 226. *See also* Amazon Web Services (AWS)
  Lazys3, 74
  S3 buckets, 61, 64, 74
Amazon Web Services (AWS), 61, 75, 308, 316
  `awscli`, 75
American Fuzzy Lop (AFL), 370
Android, 335, 347–354
  Android Debug Bridge (ADB), 351
  Android Package (APK), 350
    Activities, 350
    AndroidManifest.xml, 350
    assets, 351
    BroadcastReceivers, 350
    classes.dex, 351
    ContentProviders, 350
    lib, 351
    MANIFEST.MF, 351
    META-INF, 351
    res, 351
    resources.arsc, 351
    res/values/strings.xml, 354
    Services, 350
  Android Studio, 352
  developer options, 352
Apache
  Apache Cassandra, 199
  Apache Commons FileUpload, 243
  Apache CouchDB, 199
  Apache Groovy, 243
APIs. *See* application programming interfaces (APIs)
APK. *See* Android Package (APK)
Apktool, 352
application logic errors, 275–281, 379. *See also* business logic vulnerabilities
application programming interfaces (APIs), 6, 34, 355–367
  API-centric applications, 361
  API enumeration, 362
  API keys, 75, 226
`apt-get`, 219
ASCII, 126–127, 138–140, 293
ASNs. *See* autonomous systems (ASNs)
asset, 4. *See also* scope

attack scenarios, 19
attack surface, 5–6, 25, 61–62, 104, 309
authentication app, 276
authentication keys, 62
authorization code, 276, 314
automated testing toolkit, 379
automation strategies, 318
autonomous systems (ASNs), 67
AWS. *See* Amazon Web Services (AWS)
Ayrey, Dylan, 339

B

bash script, 62, 80–104, 372
basic authentication, 376
Big List of Naughty Strings, 372
billion laughs attack, 258. *See also* XML bomb
Bitbucket, 316
bitly.com, 119
black-box testing, 336. *See also* gray-box testing, white-box testing
blocklist, 126, 133, 215. *See also* allowlist
broken access control, 275–281, 364. *See also* access control
brute-forcing, 42, 54, 70–71, 376–377
  directory brute-forcing, 62, 70–71
  URL brute-forcing, 278
bug bounty
  bug bounty hunter, 3
  bug bounty platforms, 8
  bug bounty program, 3–4
  notes, 58
  private programs, 11
bug chains, 27
Bugcrowd, 4, 8, 17
bug slump, 27
built-in functions, 270–272, 288
BuiltWith, 79, 104
Burp, 39, 47–58
  AuthMatrix, 185
  Auto Repeater, 185
  Autorize, 185
  BAppStore, 185
  Burp Suite Pro, 47, 219
  Collaborator, 219
  comparer, 58
  crawler, 72
  decoder, 39, 57
  intruder, 54, 129, 370, 372
  repeater, 56
  SQLiPy, 203
business impact, 17, 27, 104, 379. *See also* business priorities
business logic vulnerabilities, 276. *See also* application logic errors
business priorities, 17, 27. *See also* business impact
business requirements, 279

C

CA. *See* certificate authority (CA)
capitalization, 126
CAPTCHA, 65
Capture the Flag, 12, 28
Cascading Style Sheets (CSS), 34, 147
  opacity, 148
  z-index, 147
`cat` command, 92
CDATA. *See* character data (CDATA)
Censys, 67, 70, 104
central processing units (CPUs), 206
certificate authority (CA), 50
certificate parsing, 67
certificate pinning, 349–350, 353
cert pinning. *See* certificate pinning
character data (CDATA), 259
`chmod`, 82
clickjacking, 143–154, 163–165
client, 34. *See also* server
client IDs, 313–315
Cloud computing, 226
CNAME, 308
  dangling CNAMEs, 309
Cobalt, 4, 8
Codecademy, 44, 80
code injection, 283. *See also* command injection, RCE
command injection, 285, 343. *See also* code injection, RCE
command substitution, 84, 101, 292
Common Vulnerabilities and Exposures (CVEs), 78, 281, 332, 340
Common Vulnerability Scoring System (CVSS), 17
concurrency, 206

confidentiality, 312
configuration files, 70
CORS. *See* Cross-Origin Resource Sharing (CORS)
CPUs. *See* central processing units (CPUs)
Cron, 102–103, 318
  crontabs, 102–103
Cross-Origin Resource Sharing (CORS), 297–298, 302–306
cross-site request forgery (CSRF), 128, 152, 155–174
cross-site scripting (XSS), 111–129, 308
CSRF. *See* cross-site request forgery (CSRF)
cryptography, 6–7, 339
  weak cryptography, 339
CSS. *See* Cascading Style Sheets (CSS)
CTF. *See* Capture the Flag
  CTF Wiki, 273
`curl`, 87, 211, 366
CVEs. *See* Common Vulnerabilities and Exposures (CVEs)
  CVE database, 340
CVSS. *See* Common Vulnerability Scoring System (CVSS)
CyberChef, 39
Cyrillic, 140

D

Damn Vulnerable Web Application, 203
`data:`, 122, 138
database, 188
data entry points, 371
data exfiltration, 259
data injection points, 371
debugging mode, 351
debug messages, 64
Denial-of-Service Attacks (DoS), 10, 200, 258
  ReDoS, 63
dependencies, 76, 250, 288, 340
  outdated dependencies, 76, 340
descriptive error, 196, 257, 266, 268
deserialization, 231–246
developer comments, 324, 328, 331, 340, 345
developer tools, 129
DigitalOcean, 227
directory enumerator, 370
directory traversal, 43, 177, 279, 325. *See also* path traversal
DNS. *See* Domain Name System (DNS)
DOCTYPE, 248
document.cookie, 115
Document Object Model (DOM), 117–118
document type definition (DTD), 248–250, 253–260
DOM. *See* Document Object Model (DOM)
domain name, 33. *See also* hostname
Domain Name System (DNS), 34–35
  DNS records, 222
    AAAA records, 222
    A records, 222
  DNS zone transfers, 68
domain privacy, 66
domain registrar, 65, 223
DoS. *See* Denial-of-Service Attacks (DoS)
DTD. *See* document type definition (DTD)

E

EC2. *See* Amazon Elastic Compute Cloud (EC2)
ECB, 339
`echo` command, 83
EdOverflow, 125, 317
Eloquent JavaScript, 44
embedded browser, 47, 50
emulator, 6, 348–349, 352–353
  mobile emulator, 348–349
encoding
  base64 encoding, 38, 138, 181
  content encoding, 38
  decimal encoding, 223
  double encoding, 139
  double word (dword) encoding, 223–224
  hex encoding, 38, 223
  mixed encoding, 223
  octal encoding, 223
  URL decoding, 138
  URL encoding, 38, 138, 181, 223

- encryption, 312, 338–339, 353
- entropy, 77, 159, 182, 339
- ERB. *See* Embedded Ruby template (ERB)
- escaping, 119
  - escape character, 101, 119, 293
  - output escaping, 119
- `eval`, 284–285, 336–338
- event listener, 298–300, 302–303, 305
  - `onclick`, 122
  - `onerror`, 122
  - `onload`, 122
- executable, 7
- Extensible Markup Language (XML), 247–260, 309, 357–358
  - external entities, 248
  - parameter entities, 256
  - XML entities, 248
  - XML parsers, 247
- EyeWitness, 71, 316

F

- file inclusion, 286–287
  - local file inclusions, 287
  - remote file inclusion, 286
- File Transfer Protocol (FTP), 260
- filter bypass, 128, 293
- fingerprinting, 78
- Firefox, 46–52, 124, 160–161
- Flash, 111
- Frida, 350, 353
  - Objection, 350
  - Universal Android SSL Pinning Bypass, 350
- FTP. *See* File Transfer Protocol (FTP)
- fuzzing, 125, 195, 363, 370–379
  - FuzzDB, 372
  - fuzzers, 369–379
  - web application fuzzing, 370

G

- gadgets, 238, 243–245
  - gadget chains, 243–245
- `getopts`, 92–98
- Git, 328
  - Blame, 76
  - `git diff`, 103
  - History, 76
  - Issues, 76
- GitHub, 75, 316
  - GitHub gists, 327
  - GitHub Pages, 308–309, 317
  - repositories, 75
- Gitleaks, 328
- Gitrob, 77
- Global Regular Expression Print (grep), 88–89
- GoDaddy, 219
- Google Cloud, 226–227, 316
- Google dorking, 62, 65, 74, 134, 278
- Google Hacking Database, 65
- Graphical User Interface (GUI), 373
- GraphQL, 179, 358–365
  - Clairvoyance, 362
  - introspection, 360-361
    - `__schema`, 360
    - `__type`, 361
  - mutations, 359
  - queries, 359
  - Playground, 362
- gray-box testing, 336. *See also* black-box testing, white-box testing
- grep. *See* Global Regular Expression Print (grep)
- GUI. *See* Graphical User Interface (GUI)

H

- hacker blogs, 28
- HackerOne, 4, 8, 11, 17, 111, 233
  - Hacktivity, 209
- hacking, 61
  - hacking environment, 45
- HackTricks, 273
- hardcoded secrets, 76, 338–339, 354
- hardware, 7
- hashing, 177
- Haverbeke, Marijn, 44
- HMAC, 42
- Hostinger, 219
- hostname, 67, 296. *See also* domain name
- HTML. *See* Hypertext Markup Language (HTML)
- HTTP. *See* HyperText Transfer Protocol (HTTP)
- `HttpOnly`, 115, 120

Hypertext Markup Language (HTML), 34
- HTML tag, 123

HyperText Transfer Protocol (HTTP), 36–39
- cookies, 39
  - cookie sharing, 308
  - double-submit cookie, 167
- request headers, 36
  - `Authorization`, 36, 376
  - `Cookie`, 36
  - `Host`, 36
  - `Origin`, 297
  - `Referer`, 36
  - `User-Agent`, 36, 377
- request methods, 183
- response bodies, 37, 324
- response headers, 37, 151, 324
  - `Access-Control-Allow-Origin`, 37, 297–298, 302–305
  - `Content-Security-Policy`, 37, 120, 149, 151
  - `Content-Type`, 37, 242, 251
  - `frame-ancestors`, 149
  - `Location`, 37
  - `Set-Cookie`, 37, 150, 156
  - `X-Frame-Options`, 37, 149, 151, 153–154
- response times, 9
- status code, 36, 219

I

identity assertion, 309–312

identity provider, 309–314, 316, 319

IDE. *See* integrated development environment (IDE)

IDORs. *See* insecure direct object references (IDORs)

IETF. *See* Internet Engineering Task Force (IETF)

iframe, 144–154, 158, 160, 163–164, 298–299, 304
- double iframe, 152
- `frame-busting`, 151–152

information leaks, 170, 226, 229, 295, 312, 324, 331–332, 354, 363–365

inline scripts, 113–114

input redirection, 83

input validation, 119–120, 250, 288, 291, 293, 366

insecure deserialization, 231–246, 337–338, 366–367

insecure direct object references (IDORs), 175–186, 353–354
- blind IDORs, 183
- read-based IDORs, 184
- write-based IDORs, 184

instance metadata, 226–229, 255

integrated development environment (IDE), 59

internal network, 214. *See also* private network
- internal domains, 66

internet, 33
- internet security controls, 38

Internet Engineering Task Force (IETF), 222

Internet of Things (IoT), 5, 7, 122, 347, 358

Internet Protocol (IP), 34
- IPv4, 34
- IPv6, 34, 222
- IP addresses, 65–66
  - IP range, 66
  - reserved IP addresses, 218

Intigriti, 4, 8

iOS, 348, 350, 353

IoT. *See* Internet of Things (IoT)

IP. *See* Internet Protocol (IP)

J

java.io.Serializable, 241
- `readObject()`, 241–242, 244
- `writeObject()`, 241

`javascript:`, 122–126

JavaScript (JS), 34, 44, 111, 353
- Angular, 120
- `fromCharCode()`, 126

Jenkins, 69

`jq`, 90–91

jQuery, 118
- *js.do*, 127
- React, 120
- Retire.js, 180
- Vue.js, 120

JS. *See* JavaScript (JS)
JSON, 68, 184, 234, 357
JSONP. *See* JSON with Padding (JSONP)
JSON Web Tokens (JWT), 41–43
  alg field, 42
  header, 41
JSON with Padding (JSONP), 300–302, 305–306. *See also* JSON
JWT. *See* JSON Web Tokens (JWT). *See also* JSON

K

Kali Linux, 46
KeyHacks, 76
Kibana, 64
Kubernetes, 227

L

Learn Python the Hard Way, 44
LinkFinder, 331
Linux, 62
localhost, 218
low-hanging fruit, 25

M

macOS, 62
`man`, 96
man-in-the-middle attacks, 349
Markdown, 59
Masscan, 69
MD4, 339
MD5, 339
memory leaks, 370
methodology, 25, 27
MFA. *See* multifactor authentication (MFA)
Miessler, Daniel, 372
mind-mapping, 59
mitigation process, 19–21
`mkdir`, 83
mobile applications, 6
mobile hacking, 347–354
Mobile Security Framework, 353
MongoDB, 199
monitoring system, 318
multifactor authentication (MFA), 276–277, 280
multithreading, 206
MySQL, 188, 196, 198, 201

N

Namecheap, 223
Netcat, 219
NetRange, 66
network perimeter, 214
network scanning, 215, 224–228
NoSQL, 188, 199–201
  NoSQL injections, 199–201
  NoSQLMap, 200
`nslookup`, 66, 222
NULL origin, 297–298, 303–305

O

OAuth, 141, 312–316, 320–321
  `redirect_uri`, 313–316
object-oriented programming languages, 234
Obsidian, 59
Offensive Security, 120
open redirect, 131–141, 221, 314–316, 338, 342–343
  open redirect chain, 315
  parameter-based open redirects, 135
  referer-based open redirects, 132, 135
operating system, 46, 62
OSINT, 77
outbound requests, 228, 249, 252
out-of-band interaction, 289
out-of-band techniques, 219
output redirection, 83–84
OWASP, 28, 72
  Code Review Guide, 336
  Dependency-Check tool, 340
  Deserialization Cheat Sheet, 244
  IoTGoat, 122
  Mobile Security Testing Guide, 348
  SQL injection prevention cheat sheet, 195
  Web Security Testing Guide, 367
  XSS filter evasion cheat sheet, 128
  XSS prevention cheat sheet, 120

P

- parameterized queries, 192. *See also* prepared statements
- parent directory, 279, 325
- passive scanning, 69–70. *See also* active scanning
- password-cracking, 269
- Pastebin, 77–78, 324, 327–328
  - pastebin-scraper, 328
  - PasteHunter, 78, 328
- paste dump sites, 327
- path enumeration, 374–375
- path traversal, 177, 279, 325, 366–367. *See also* directory traversal
- `PATH` variable, 81
- pattern matching, 89
- payload, 41, 54, 154
- payouts, 9–11
- Periscope, 153
- permissions, 178
- permutations, 69, 74–75
- phishing, 129, 132, 140, 309
- PHP, 61, 70–71, 232–241
  - ExtendsClass, 232
  - instantiation, 235, 239
  - magic methods, 235–238
  - object injection vulnerabilities, 233, 238
    - `unserialize()`, 235
  - wrappers, 259
- *phpmyadmin*, 70, 79
- `PHPSESSID`, 79
- POC. *See* proof of concept
- POP chain. *See* property-oriented programming chain
- pop-up, 154
- port, 35
  - port number, 35, 296
  - port scanning, 62, 69
- Postman, 362
- `postMessage()`, 298–306
- prepared statements, 192–194
- principle of least privilege, 201, 210, 288
- private network, 218. *See also* internal network
- Programmer Help, 273
- programming, 44
  - expression, 262
  - `for` loop, 93
  - function library, 96
  - functions, 87
  - `if-else` statements, 86
  - interactive programs, 97
  - statement, 262
  - `while` loop, 98
- Project Sonar, 70
- proof of concept (POC), 18
  - POC generation, 174
- property-oriented programming chain, 238–239
- protocol, 43, 120, 296, 325
- proxy, 46, 52, 72, 348
  - proxy services, 216
  - web proxy, 45
- publicly disclosed reports, 25. *See also* write-up
- publicly disclosed vulnerabilities, 324
- Python, 44, 244–245, 262–273, 289–292
  - dictionary, 272
  - object, 270

Q

- Quora, 77

R

- race conditions, 205–212, 366, 370
- randomization, 178
- rate-limiting, 365–366, 378
- RCE. *See* remote code execution (RCE)
- reachable machines, 224
- recon. *See* reconnaissance
- reconnaissance, 25, 61–107, 243, 360, 369
  - recon APIs, 104
- referer, 132–135, 141–163, 168–169, 315
- regex. *See* regular expression
- regular expression, 77, 88–90, 221, 298, 338–339
  - constants, 89
  - operators, 89
  - RexEgg, 90
- remote code execution (RCE), 236–237, 283–293, 337
  - blind RCEs, 288
  - classic RCEs, 288

report states, 21
- duplicate, 22
- informative, 22
- invalid reports, 26
- low-severity bug, 26
- mediation, 23
- N/A, 22
- need more information, 22
- resolved, 23
- triaged, 22

Representational State Transfer (REST), 357
resource locks, 210
REST. *See* Representational State Transfer (REST)
return-oriented programming, 241
reverse engineering, 6
reverse shell, 285
rooted device, 6
RSA, 42

S

S3. *See* Amazon S3
safe concurrency, 206
same-origin policy (SOP), 43, 295–306
SameSite, 149–152, 159–160
SAML. *See* Security Assertion Markup Language (SAML)
sandbox
- sandbox environment, 265–166
- sandbox escape, 269–273

sanitizing, 114
SAST. *See* static analysis security testing (SAST)
SCA. *See* software composition analysis (SCA)
scanner, 72
scheduling, 206
scope, 9–13, 26
- scope discovery, 65

search engine, 63
SecLists, 68, 372
secret-bridge, 325
secret key, 40
secret storage system, 325
Secure Shell Protocol (SSH), 218, 225, 227
Secure Sockets Layer (SSL), 67, 349
Security Assertion Markup Language (SAML), 309
- SAML Raider, 320
- SAML signature, 311

security context, 302
security patches, 340
security program, 4
sensitive data leaks, 312
sensitive information, 324
serialization, 232. *See also* deserialization
- serialized string, 233

server, 34, 79. *See also* client
- server logs, 64
- server status, 64

server-side request forgery (SSRF), 213–229, 278
- blind SSRF, 214

server-side template injections (SSTIs), 261–274
server-side vulnerabilities, 6
service banner, 218
service enumeration, 69
service provider, 309
session, 39–40
- session cookie, 115, 156–160, 162–172, 308–309, 318–321. *See also* session ID
- session ID, 39. *See also* session cookie
- session management, 39

Shaw, Zed, 44
shebang, 80
shell
- commands, 285
- interpreter, 62

Shopify, 359
signature, 40–43, 311–312, 319–321, 351
single sign-on (SSO), 307–321
- shared-session SSO, 308–309

SlideShare, 77
Snapper, 71, 316
SOAP, 358
social engineering, 119, 132
- Social-Engineer Toolkit, 154

software composition analysis (SCA), 340
software supply chain attack, 288

SOP. *See* same-origin policy (SOP)
  source code review, 76, 328, 335–346, 351, 378. *See also* static analysis security testing (SAST), static code analysis
`source` command, 96
spidering, 62, 71
Spring Framework, 243
SQL. *See* Structured Query Language (SQL)
SQL injections, 187–203
  blind, 188, 195
  Boolean based, 196
  classic, 188, 195
  error based, 195
  first-order, 191
  inferential, 196
  out-of-band, 188, 195
  second-order, 191
  time based, 197
  UNION based, 195
sqlmap, 202
Squarespace, 316
SSH. *See* Secure Shell Protocol (SSH)
SSL. *See* Secure Sockets Layer (SSL)
SSL pinning. *See* certificate pinning
SSO. *See* single sign-on (SSO)
SSRF. *See* server-side request forgery (SSRF)
SSRFmap, 220
SSTIs. *See* server-side template injections (SSTIs)
Stack Overflow, 77
state-changing action, 149, 161
static analysis security testing (SAST), 346. *See also* source code review, static code analysis
static code analysis, 378. *See also* source code review, static analysis security testing (SAST)
Structured Query Language (SQL), 187–188
SubBrute, 68
subdomain, 64–65
  sibling subdomains, 296
  subdomain enumeration, 68–69, 379
  subdomain takeovers, 308–309, 316–318
Subject Alternative Name, 67–68
Sublime Text, 59
superdomain, 296
SVG, 253
Swagger, 363
Synack, 4, 8
synchronization, 210
syntax error, 123
system root, 279

T

technology stack, 6, 69, 78–79, 104
template engines, 261–266
  Embedded Ruby template (ERB), 266
  FreeMarker, 266
  Jinja, 262
  Smarty, 266
  Thymeleaf, 266
  Twig, 266
template injections. *See* server-side template injections (SSTIs)
`test` command, 95
testing guides, 28
third-party service, 308
threads, 206
time-of-check/time-of-use vulnerabilities. *See* race conditions
time throttling, 366, 373–374
token-based authentication, 40
token forgery, 40
Tomnomnom, 78
tplmap, 273
triage, 8–9
truffleHog, 77, 328, 339
tuple, 270
Tutorials Point, 232
Twitter, 356

U

Unarchiver, 253
unexpected behavior, 370
Unicode, 140
Unix, 46, 81, 100–102, 177, 249, 279, 290, 292, 325, 372
Unrouted addresses, 228
URLs, 63
  absolute URL, 133–134, 325

components of, 136
internal URLs, 218
mangled URLs, 136
relative URLs, 133, 325
URL fragments, 118, 121, 266
URL validation, 133, 136
USB debugging, 351
user input, 342
user-interface redressing, 143. *See also* clickjacking

V

validating, 114
Vault, 325
VBScript, 111
VDPs. *See* vulnerability disclosure programs (VDPs)
ViewDNS.info, 66
View Source Code, 79
virtual environment, 352
vulnerabilities, 61
vulnerability disclosure programs (VDPs), 10
vulnerability report, 16. *See also* write-up
severity, 16
steps to reproduce, 18
vulnerability scanners, 25

W

W3Schools, 188
WAF. *See* web application firewall (WAF)
Wappalyzer, 79
Wayback Machine, 326
Waybackurls, 78
web application firewall (WAF), 288
WAF bypass, 293
web applications, 5
web browser, 46
web crawling, 71, 326
web frameworks, 187
Webhooks, 216
web-hosting service, 223
web page, 34
Web Services Description Language (WSDL), 358, 362
web shell, 202
web spidering, 62, 71
Wfuzz, 370–371, 374–379
`wget`, 285, 329
white-box testing, 336. *See also* black-box testing, gray-box testing
`whoami`, 289
`whois`, 65
reverse whois, 65
whois.cymru.com, 67
Wikipedia, 63
wildcard, 63, 101, 292, 297–299
Windows 353
WordPress, 7, 79, 280
write-up, 28
WSDL. *See* Web Services Description Language (WSDL)

X

XInclude Attacks, 251, 254
XMind, 59
XML. *See* Extensible Markup Language (XML)
XML bomb, 258. *See also* billion laughs attack
XML external entity (XXE), 247–260
blind XXE, 252
classic XXE, 251
`XMLHttpRequest`, 128, 170
XmlLint, 259
`X-Powered-By`, 79, 324
XSS, 111–129
blind XSS, 116, 125
reflected XSS, 117, 343
self-XSS, 119, 171
stored XSS, 115
XSS filter, 126
XSS Hunter, 125
XSS polyglot, 124
XSS protection, 126
XXE. *See* XML external entity (XXE)

Y

YAML, 234, 338
Ysoserial, 243

Z

ZAP. *See* Zed Attack Proxy (ZAP)
Zed Attack Proxy (ZAP), 47, 72–73, 174, 362, 374
`zip` command, 254
zlib, 331