

# INDEX

Page numbers followed by an italicized *f* or *t* refer to figures and tables respectively.

## Symbols and Numbers

: (colon character), 82  
/ (path separator character), 109  
; (semicolon character), 52  
' (single quote character), 51–53  
10 Minute Mail, 86  
401 status code (in HTTP), 82

## A

Accept header, 10–11  
access control, 25, 166  
    aspects of, 104  
    audit trails, 107–108  
    common oversights, 108  
    defined, 104  
    implementing, 106–107  
    models for  
        access control lists, 105  
        ownership-based access control, 106  
        role-based access control, 105–106  
        whitelists and blacklists, 105  
    testing, 107  
access control lists (ACLs), 105  
access tokens, 139  
Acid3 test, 18, 18*f*  
Active Directory, 137  
ActiveRecord framework, 54–55  
ActiveX, *xxi*  
administrative frontends, securing, 138  
ad platforms, 142  
Advanced Encryption Standard (AES), 121, 138  
Advanced Research Projects Agency Network (ARPANET), 7

Airbrake, 44  
Akamai, 26, 62, 138  
Amazon  
    denial-of-service attacks, 163  
    one-click purchases, 76  
Amazon CloudFront, 26  
Amazon Elastic Compute Cloud (EC2), 41  
Amazon Machine Images (AMIs), 62  
Amazon Simple Storage Service (S3), 62, 140  
Amazon Web Services (AWS), 105, 137, 168  
    Elastic Beanstalk, 41  
    Marketplace, 62  
AMIs (Amazon Machine Images), 62  
amplified attacks, 165  
Angular framework, 33–34, 72  
Ansible, 42  
anti-CSRF cookies, 77–78  
antivirus software  
    mitigating file upload vulnerability attacks, 63  
    protection against botnets, 160  
Apache web servers, 53, 114, 125  
    disabling open directory listings, 137  
    disabling URL rewriting, 100  
    injection attacks, 132  
application firewalls, 166  
application layer attacks, 165  
application programming interface (API) keys, 139  
application servers, 125–126  
ARPANET (Advanced Research Projects Agency Network), 7  
asymmetric encryption algorithms, 119  
audit trails, 107–108  
authentication  
    brute-force attacks, 83–84  
    databases and, 29  
    defined, 81

- authentication (*continued*)
  - implementing
    - basic authentication
      - scheme, 82
    - digest authentication
      - scheme, 82
    - HTTP-native authentication,
      - 82–83, 83*f*
    - non-native authentication, 83
  - mitigation options
    - secure authentication system,
      - 85–92
    - single sign-on, 84–85
    - third-party authentication, 84
- authenticator apps, 90, 90*f*
- Authorization header, 82
- AVG, 132
- AWS. *See* Amazon Web Services (AWS)

## B

- Bachus-Naur Form (BNF), 147
- Base64 algorithm, 82
- bcrypt algorithm, 88–89
- Berners-Lee, Tim, xx–xxi, 24
- bind parameters
  - object-relational mapping, 54
  - parameterized statements, 52–53
- Bitly, 156
- black hat hackers, 2
- blacklists, 105
- blind SQL injection attacks, 56
- block ciphers, 119–120
- BNF (Bachus-Naur Form), 147
- botnets, xxii, 154, 160, 165
- branching code, 38
- brittleness, 39
- browsers
  - cookies, 20
  - Document Object Model, 16–17
  - Domain Name System, 20
  - HTTP Secure, 20
  - JavaScript, 16, 18–19
  - security certificates, 20
  - styling rules, 16–18
  - web page rendering pipeline, 15–19
- brute-force attacks, 100
- bug trackers (issue-tracking software), 36
- building for scale, 167, 173
- bundler-audit tool, 136

## C

- C#
  - build process, 42
  - overview of, 33
  - vulnerabilities, 33
- C++, 33
- Cache-Control header, 13
- canonical name (CNAME) records, 9
- CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), 91–92, 92*f*, 158
- Cascading Style Sheets (CSS)
  - build process, 43
  - in HTTP responses, 13
  - pre-processors, 43
  - selectors, 17
  - stylesheets, 17
  - styling rules, 17
  - use in clickjacking attacks, 158
- CDNs. *See* content delivery networks (CDNs)
- Center for Internet Security (CIS), 62
- centralized version control systems, 37
- Centrify, 84
- CEO fraud, 154
- CERN (European Organization for Nuclear Research), xx–xxi
- certificate authorities, 117, 122–125
- certificate signing requests (CSRs), 123–124
- CGI (Common Gateway Interface), xxii
- checksums, 8, 40, 134
- Chef, 42
- chroot command, 58
- CIS (Center for Internet Security), 62
- Cisco, 128
- cleartext storage, 88
- click fraud, 160
- clickjacking, 154, 158–159
- client-server architecture, 49
- client-side error reporting, 115
- client-side sessions, 96–97
- Clojure, 32
- cloud-based storage
  - hosting services, 110–111
  - subdomain takeovers, 140
  - use in mitigating file upload vulnerability attacks, 62
- Cloudflare, 26, 62
- CLR (Common Language Runtime), 33

- CMSs. *See* content management systems (CMSs)
- CNAME (canonical name) records, 9
- code reviews, 38, 170
- code writing phase (in the software development lifecycle)
  - branching and merging code, 38
  - pushing changes to repository, 37
  - source control (version control), 37
- CoffeeScript, 34, 42
- colon character (:), 82
- Comcast, 128
- command injection attacks
  - anatomy of, 56–57, 57*f*
  - defined, 56
  - escaping control characters, 57–58
  - file upload vulnerability and, 61
- Common Gateway Interface (CGI), xxii
- Common Language Runtime (CLR), 33
- Comodo, 122
- Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA), 91–92, 92*f*, 158
- CONNECT requests (in TCP), 11*t*
- consistent behavior
  - eventual consistency of NoSQL databases, 30
  - SQL databases, 29
- containerization, 42
- content delivery networks (CDNs), 26
  - distributed denial-of-service protect systems, 167
  - mitigating file upload vulnerability attacks, 62, 170
  - subdomain takeovers, 140
- content management systems (CMSs)
  - defined, 26
  - mitigating file upload vulnerability attacks, 62, 170
  - plug-ins, 26
  - vulnerabilities of, 26, 27*f*
- content security policies, 69–70, 110–111
- Content-Security-Policy header, 69, 158–159, 172
- Content-Type header, 13, 59, 63
- continuous integration servers, 39
- control characters
  - in PHP, 56–58
  - in SQL, 51–53, 55
- Cookie header, 13, 77–78, 97
- cookies, 171
  - anti-CSRF cookies, 77–78
  - defined, 13
  - digital signing of, 96–97
  - generic cookie parameters, 114
  - implementing and securing a
    - logout function, 90–91
  - SameSite cookie attribute, 78–79
  - session cookies, 95–97, 114
  - session hijacking, 97–99
  - vulnerabilities of, 13
- cookie theft
  - cross-site request forgery (CSRF) attacks, 98–99
  - cross-site scripting (XSS) attacks, 97–98
    - man-in-the-middle attacks, 98
- cracking password lists, 89
- CREATE statements (in SQL), 55
- cross-site request forgery (CSRF; XSRF) attacks
  - anatomy of, 76
  - cookie theft, 98–99
  - defined, 75
  - mitigation options
    - anti-CSRF cookies, 77–78
    - requiring reauthentication for sensitive actions, 78–79
    - REST principles, 76–77
    - SameSite cookie attribute, 78–79
- cross-site scripting (XSS) attacks, xxi, 19
  - cookie theft, 97–98
  - defined, 65
  - DOM-based
    - defined, 71
    - escaping dynamic content, 73
    - URI fragments, 71–73
  - reflected, 70–71
    - defined, 70
    - escaping dynamic content, 71
  - stored
    - content security policies, 69–70
    - escaping control characters, 67–69
    - example of, 66–67, 66*f*–67*f*
- cryptographic hash algorithms and functions, 88–89, 119
- cryptology, 117

CSRF attacks. *See* cross-site request forgery (CSRF) attacks  
CSRs (certificate signing requests), 123–124  
CSS. *See* Cascading Style Sheets (CSS)

## D

database administrators (DBAs), 43  
database drivers, 51  
database migration scripts, 43  
databases  
    authentication and, 29  
    for building dynamic web pages, 28–30  
    NoSQL, 30  
    origin of, 28  
    SQL, 29–30, 105  
    stored cross-site scripting attacks, 66  
data definition language (DDL), 55  
data integrity, 29, 118  
data manipulation language (DML), 55  
data packets, 8  
DBAs (database administrators), 43  
DDL (data definition language), 55  
DDoS (distributed denial-of-service) attacks, 165, 167  
decryption keys, 118–119  
dedicated configuration stores, 137  
default credentials, disabling, 137  
defense in depth approach, 61  
    blind and nonblind SQL injection, 55–56  
    defined, 55  
    principle of least privilege, 55  
defer attribute, 19  
DELETE requests (in SQL), 11, 76–77  
DELETE statements (in SQL), 50–51, 55  
denial-of-service (DoS) attacks  
    defined, 163–164  
    mitigation options  
        building for scale, 167–168  
        firewalls, 166  
        intrusion prevention systems, 166  
        protection services, 167  
    types of  
        application layer attacks, 165  
        distributed denial-of-service attacks, 165  
        Internet Control Message Protocol (ICMP) attacks, 164

    reflected and amplified attacks, 165  
    Transmission Control Protocol (TCP) attacks, 164  
    unintentional denial-of-service attacks, 166  
dependencies, 171  
    build process, 42  
    defined, 45  
    deploying new versions quickly, 134  
    organizing dependencies  
        dependency management tools, 132–133  
        operating system patches, 133–134  
        subresource integrity checks, 134  
    security advisories  
        blogs, 135  
        mailing lists, 135  
        official advisories, 135  
        social media, 135  
        software tools, 136  
        timely upgrades, 136  
        vulnerability of, 45  
dependency-check tool, 136  
dependency management, 45, 132–133  
dependency trees, 133  
deserialization  
    defined, 59  
    disabling code-execution during, 59–60  
design and analysis phase, 36  
DevOps (developer operations) tools, 41–42  
DigiCert, 122  
digital certificates (public-key certificates)  
    certificate authorities, 122–123  
    defined, 122  
    installing  
        configuring web server to use HTTPS, 126  
        HTTP Strict Transport Security policies, 127  
        redirecting HTTP traffic to HTTPS, 126–127  
        web servers vs. application servers, 125–126  
obtaining  
    certificate signing requests, 123–124  
    domain verification, 124

- expiring certificates, 124
    - extended validation
      - certificates, 124
    - key pairs, 123–124
    - paying for certificates, 125
    - revoking certificates, 124
    - self-signed certificates, 124–125
  - TLS handshakes, 121
  - digital signatures, 96–97
  - directories, 109
  - directory traversal attacks, xxii, 108–112
    - anatomy of, 109–110, 109*f*–110*f*
    - defined, 108
    - filepaths and relative filepaths, 108–109
    - mitigation options, 171
      - hosting services, 110–111
      - indirect file references, 111
      - sanitizing file references, 111–112
      - web servers, 110
  - display names, 85
  - distinguished names (DNs), 123
  - distributed caches
    - defined, 30
    - injection attacks, 53
    - microservices, 30
    - publish-subscribe channels, 31
    - queues, 30
    - vulnerabilities of, 31
  - distributed denial-of-service (DDoS)
    - attacks, 165, 167
  - distributed version control systems, 37
  - Django, 125
  - DKIM (DomainKeys Identified Mail), 155–156, 172
  - DML (data manipulation language), 55
  - DNS. *See* Domain Name System (DNS)
  - DNs (distinguished names), 123
  - DNS poisoning, 9
  - Docker, 42, 62, 134
  - Docker Swarm, 42
  - Document Object Model (DOM)
    - defined, 16
    - DOM-based cross-site scripting
      - attacks, 71–73
    - DOM nodes, 17
    - DOM tree, 17
    - HTML tags and, 17
    - rendering pipeline, 16–17
  - document type definition (DTD) files, 147–150
  - DomainKeys Identified Mail (DKIM), 155–156, 172
  - domain name servers, 9
  - Domain Name System (DNS)
    - caching behavior, 9
    - canonical name records, 9
    - DNS poisoning, 9
    - domain verification, 124
    - encryption, 122–123
    - Internet Protocol suite layers, 10*f*
    - mail exchange records, 9
    - purpose of, 9
    - registration of, 9
    - rendering pipeline, 20
    - subdomain takeovers, 140
    - validating email addresses, 86
  - domain registrars, 9
  - domain verification, 124
  - DOM-based cross-site scripting attacks
    - defined, 71
    - escaping dynamic content, 73
      - URI fragments, 71–73
  - doppelganger domains, 154
  - DoS attacks. *See* denial-of-service (DoS) attacks
  - downgrade attacks, 128
  - DROP command and statements
    - (in SQL), 52, 55
  - DTD (document type definition) files, 147–150
  - Dyn, 163, 166
  - dynamic resources
    - databases, 28–30
    - defined, 24
    - distributed caches, 30–31
    - templates, 28
    - web programming languages, 31–34

## E

  - EC2 (Amazon Elastic Compute Cloud), 41
  - Eich, Brendan, xxi
  - Electronic Frontier Foundation, 125
  - Elliptic Curve Diffie-Hellman Exchange (ECDHE), 121
  - email addresses
    - banning disposable, 86, 87*f*
    - requiring for authentication, 85
    - spoofing, 153
    - validating, 85–86

- email fraud
    - avoiding
      - DomainKeys Identified Mail, 155–156, 172
      - Sender Policy Framework, 155–156, 171
    - email address spoofing, 153
    - open redirects, 156–157, 172
    - phishing, 154
    - spam, 154
    - spearphishing, 154
  - email verification links, 86
  - embargoing resources, 108
  - Embedded Ruby (ERB) templates, 68
  - encoded separator characters, 111
  - encryption, 171
    - algorithms
      - asymmetric, 119
      - decryption keys, 118–119
      - defined, 118
      - symmetric, 119
    - encrypting session cookies, 96
    - exploiting unencrypted communication
      - government agencies, 129
      - Internet service providers, 128–129
      - Wi-Fi hotspots, 128
      - wireless routers, 128
    - of configuration information, 138
    - handshakes, 14
    - HTTPS, 14
      - defined, 118
      - digital certificates, 117, 122–123
      - installing certificates, 125–127
      - obtaining certificates, 123–125
    - in the Internet Protocol
      - encryption algorithms, 118–119
      - hash functions, 119–120
      - message authentication codes, 120
      - TLS handshakes, 120–122
      - Transport Layer Security (TLS), 14
  - entity encodings (in HTML), 67–68, 68*t*
  - enumeration of users
    - CAPTCHA, 91–92, 92*f*
    - error messages, 91
    - password-reset screen, 91
    - timing attacks, 91
  - environmental variables, 137
  - ERB (Embedded Ruby) templates, 68
  - error reporting
    - client-side, 115, 171
    - defined, 44
    - third-party services, 44–45
  - escapeshellarg function (in PHP), 58
  - escaping control characters
    - in dynamic content from HTTP requests, 70
    - in dynamic content from URI fragments, 73
    - in HTML, 67–69, 170
    - in PHP, 57–58
    - in SQL, 53
  - European Organization for Nuclear Research (CERN), xx–xxi
  - EV (extended validation)
    - certificates, 124
  - executable files, 60–61
  - exploit kits, 141–142
  - exploits
    - defined, 1
    - white hat *vs.* black hat hackers, 2
    - zero-day exploits, 2
  - exploit scripts, 59
  - extended validation (EV)
    - certificates, 124
  - Extensible Markup Language (XML)
    - defined, 145
    - document type definition files, 147–148
    - external entity attacks, 149–150
    - parsing, 146–150, 171
    - securing XML parsers, 150–151
    - server-side request forgery attacks, 159
    - uses for, 146
    - validating, 147–148
    - XML bombs, 148–149
  - Extensible Messaging and Presence Protocol (XMPP)
    - defined, 10
    - Internet Protocol suite layers, 10*f*
  - external entity declarations, 149–150
- ## F
- Facebook, xxii
    - likejacking, 158
    - React framework, 34
    - user permissions failure, 103–104
  - Facebook Login, 26, 84, 138
  - filepaths, 108–109

- File Transfer Protocol (FTP)
  - defined, 10
  - Internet Protocol suite layers, 10*f*
- file upload vulnerability attacks
  - anatomy of, 60–61, 61*f*
  - defined, 60
  - file upload functions, defined, 60
  - mitigation options, 61–63, 170
    - ensuring uploaded files cannot be executed, 62
    - hosting files on secure system, 62
    - running antivirus software, 63
    - validating content of uploaded files, 63
- firewalls, 166
- Flask, 125
- foreign keys (in SQL), 29
- four eyes principle, 38
- FTP. *See* File Transfer Protocol (FTP)
- fully qualified domain names (FQDNs), 123

## G

- Galois/Counter Mode (GCM), 121
- GET requests (in HTTP), 11, 49–50
  - cross-site request forgery attacks, 76–77
  - rendering pipeline, 20
  - SameSite attribute settings for cookies, 99
- Git, 37–38
- GitHub, 37–38, 136, 163, 165
- GitHub OAuth, 84
- Google
  - Angular framework, 33–34
  - government snooping, 129
  - HTTP requests, 70
  - reCAPTCHA widget, 92
  - returning dynamic resources, 27
- Google AdSense, 138, 141–143
- Google AdX, 142
- Google Analytics, 26, 140–141, 168
- Google App Engine, 41
- Google Apps, 70
- Google Chrome, xxi, 83*f*
  - cipher suites, 121
  - V8 JavaScript engine, 32
- Google Hacking Database, 136
- Google OAuth, 84
- Google Safe Browsing API, 158

- government agencies, snooping by, 129
- gzip algorithm, 25

## H

- Hacker News, 135
- hacking
  - black hat hackers, 2
  - dark web, 2, 2*f*
  - exploits, defined, 1
  - process for, 3–4
  - white hat hackers, 2
  - zero-day exploits, 2
- hardening servers, 62
- hashed values, 88, 119–120
- hashes, 171
  - digest authentication scheme, 82
  - hashing passwords, 88, 119–120
  - salting hashes, 89, 171
- headers
  - in HTTP requests, 10–11
  - in HTTP responses, 12–13, 25
- HEAD requests (in HTTP), 11*t*
- Heartbleed bug, 132
- Heartland Payment Systems, 50
- Heroku, 41, 168
- horizontal escalation, 104
- hosting services, 110–111
- HSTS (HTTP Strict Transport Security) policies, 126–127
- HTML. *See* HyperText Markup Language (HTML)
- HTTP. *See* HyperText Transfer Protocol (HTTP)
- HTTP 404 Not Found error, 13
- HttpOnly keyword, 97–98, 171
- HTTP requests, 170–171
  - authentication, 82
  - command injection attacks, 56–58
  - CONNECT requests, 11*t*
  - cross-site request forgery attacks, 76
  - defined, 10
  - DELETE requests, 11
  - elements of
    - body, 10
    - headers, 10–11
    - methods (verbs), 10–11
    - universal resource locators, 10–11
  - example of, 10–11
  - exploit scripts, 59

- HTTP requests (*continued*)
  - file upload vulnerability attacks,
    - 60, 63
  - GET requests, 11
  - HEAD requests, 11*t*
  - logging, 44
  - OPTIONS requests, 11*t*
  - PATCH requests, 11
  - POST requests, 11
  - privilege escalation, 104
  - PUT requests, 11
  - reflected cross-site scripting attacks, 70–71
  - server-side request forgery attacks, 159–160
  - static resources, 24
  - TRACE requests, 11*t*
  - use in SQL injection attacks, 51–52
- HTTP responses
  - authentication, 82
  - defined, 10
  - disabling telltale headers, 114
  - elements of
    - body, 12–13
    - headers, 12–13
    - status codes, 12–13
    - status messages, 12
  - example of, 12
  - HTML, 13
  - monitoring, 44
  - returning dynamic resources, 27
  - returning static resources, 25
  - static resources, 24
- HTTP Secure (HTTPS), 14
  - cookie theft, 98
  - defined, 118
  - digital certificates
    - defined, 122
    - installing, 125–127
    - obtaining, 123–125
  - redirecting HTTP traffic to, 126–127
  - rendering pipeline, 20
  - terminating, 126
  - vulnerabilities avoided by using, 128–129
- HTTP sessions
  - cross-site scripting attacks, 65
  - defined, 13
  - implementing
    - client-side sessions, 96–97
    - server-side sessions, 94–95
  - opening, 94
  - session cookies, 95–96
  - session hijacking, 93–101
  - session IDs, 94–95
  - session state, 94–95
  - tracking, 13
  - vulnerability of, 13
- HTTP Strict Transport Security (HSTS) policies, 126–127
- hug of death, 166
- HyperText Markup Language (HTML)
  - dynamic page creation, xxii
  - in HTTP responses, 12–13
  - origin of, xxi
  - rendering pipeline, 15–17
  - tags, 17, 67–68, 68*t*
  - web servers, 23–24, 27–28
- HyperText Transfer Protocol (HTTP)
  - authentication, 82–83
  - defined, 10
  - encryption, 14
  - HTTP requests
    - CONNECT requests, 11*t*
    - defined, 10
    - DELETE requests, 11
    - elements of, 10
    - example of, 10–11
    - GET requests, 11
    - HEAD requests, 11*t*
    - OPTIONS requests, 11*t*
    - PATCH requests, 11
    - POST requests, 11
    - PUT requests, 11
    - TRACE requests, 11*t*
  - HTTP responses
    - defined, 10
    - elements of, 12
    - example of, 12
    - HTML, 13
    - HTTP headers, 12–13
    - status codes, 12–13
    - status messages, 12
- Internet Protocol suite layers, 10*f*
  - origin of, xxi
  - purpose of, 10
  - redirecting traffic to HTTPS, 126–127
  - rendering pipeline, 20
  - stateful connections, 13
  - user agents, 10



- vulnerabilities of
  - government agencies, 129
  - Internet service providers, 128–129
  - Wi-Fi hotspots, 128
  - wireless routers, 128
  - web servers, 23–25, 27

## I

- IaaS (Infrastructure as a Service), 41, 168
- ICANN (Internet Corporation for Assigned Names and Numbers), 8
- ICMP (Internet Control Message Protocol) attacks, 164
- identity and access management (IAM) system, 105
- `<iframe>` tags, 142, 158, 172
- images (configuration scripts), 42
- indirection, 111
- infinite scrolling, 72
- information leaks, 171
  - mitigation options, 113–116
    - disabling client-side error reporting, 115
    - disabling telltale Server headers, 114
    - minifying or obfuscating JavaScript files, 115
    - sanitizing client-side files, 116
    - use generic cookie parameters, 114
    - using clean URLs, 114
  - security advisories, 116
  - zero-day vulnerabilities, 112
- Infrastructure as a Service (IaaS), 41, 168
- injection attacks, xxii
  - anticipating, 170
  - client-server vulnerabilities, 49–50
  - command injection attacks
    - anatomy of, 56–57, 57*f*
    - defined, 56
    - escaping control characters, 57–58
  - defined, 49
  - file upload vulnerability attacks
    - anatomy of, 60–61, 61*f*
    - defined, 60
    - ensuring uploaded files cannot be executed, 62
    - hosting files on secure system, 62

- running antivirus software, 63
- validating content of uploaded files, 63
- remote code execution attacks
  - anatomy of, 59
  - defined, 59
  - disabling code-execution during deserialization, 59–60
- SQL injection attacks
  - anatomy of, 51–52
  - defense in depth, 55–56
  - object-relational mapping, 54–55
  - parameterized statements, 52–53
  - SQL, defined, 50–51
- INSERT statements (in SQL), 50–51, 55
- integration testing, 39
- integrity checkers, 134, 160
- internal entity declarations, 148
- internet, history of, xx–xxiii
- Internet Control Message Protocol (ICMP) attacks, 164
- Internet Corporation for Assigned Names and Numbers (ICANN), 8
- Internet Protocol (IP)
  - encryption algorithms, 118–119
  - hash functions, 119–120
  - message authentication codes, 120
  - TLS handshakes, 120–122
- Internet Protocol (IP) addresses
  - allotment of, 8
  - defined, 8
  - IP version 4 (IPv4) syntax, 8–9
  - IP version 6 (IPv6) syntax, 9
  - rendering pipeline, 20
- Internet Protocol suite
  - defined, 8–9
  - Domain Name System, 9
  - HyperText Transfer Protocol, 10–13
    - encryption, 14
    - stateful connections, 13
  - Internet Protocol addresses, 8–9
  - layers of, 9–10, 10*f*
  - Transmission Control Protocol, 8
  - User Datagram Protocol, 8
- Internet service providers (ISPs), 8, 128–129
- Intrusion prevention systems (IPSs), 166

IP. *See* Internet Protocol (IP); Internet Protocol (IP) addresses  
IP version 4 (IPv4) syntax, 8–9, 10*f*  
IP version 6 (IPv6) syntax, 9, 10*f*  
ISPs (Internet service providers), 8, 128–129  
issue-tracking software (bug trackers), 36

## J

### Java

- application servers, 125
- build process, 42
- command injection attacks, 56
- dependency checker, 136
- overview of, 32
- securing XML parsers, 151

### JavaScript

- build process, 42–43
- client-side, 33–34
- Comcast advertisements in HTTP traffic, 128
- cookie theft, 97
- cross-site request forgery attacks, 77–78
- cross-site scripting attacks, 65–73
- defined, 18
- file upload functions, 60–62
- in HTTP responses, 13
- <iframe> tags, 142
- inline, preventing execution of, 69
- minifying files, 42–43, 115
- Node.js, 32
- obfuscating files, 115
- origin of, *xxi*
- password complexity ratings, 88
- rendering pipeline, 16, 18–19
- sandboxing, 19, 141
- V8 JavaScript engine, 32
- vulnerability of, 19

### JavaScript Object Notation (JSON)

- NoSQL databases, 30
- session state, 36
- XML vs., 146

### JavaScript XML (JSX) files, 34

### Java Servlet Specification, 99

### Java Virtual Machine (JVM), 32–33

### job queues, 167

### JSESSIONID cookie, 114

### JSON. *See* JavaScript Object Notation (JSON)

JSX (JavaScript XML) files, 34  
JVM (Java Virtual Machine), 32–33

## K

### Kali Linux, 3–4

key-exchange algorithm, 121

key pairs, 123–124

key-value storage, 30

Kotlin, 33

Krebs, Brian, 135

Kubernetes, 42

## L

Let's Encrypt, 122, 125

Lightweight Directory Access Protocol (LDAP), 53

likejacking, 158

LinkedIn, 84

Linksys, 128

Linux, *xiv*

- chroot command, 58

- filepaths, 108

- file permissions, 62

- filesystem, 105

- Kali Linux, 3–4

- restricting web server-accessible directories, 152

- wireless routers, 128

Lisp, 32

logging, 44–45

## M

MACs (message authentication codes), 120

Mailchimp, 156

mail exchange (MX) records, 9, 86

Mailgun, 156

Mailinator, 86

malvertising, 141

malware, *xiii*, 141–142

man-in-the-middle attacks, 14

- cookie theft, 98

- by government agencies, 129

- unsecured TCP conversations, 118

- Wi-Fi hotspots, 128

- wireless routers, 128

Memcached, 30–31, 53, 165

merge conflicts, 38

merging code, 38

- message authentication codes
  - (MACs), 120
- Metasploit framework, 3–4, 3*f*, 52, 114
- <meta> tags, 69
- methods (verbs) in HTTP, 10–11
- MFA. *See* multifactor authentication (MFA)
- microframeworks, 31
- microservices
  - defined, 30
  - distributed caches, 30
  - publish-subscribe channels, 31
  - queues, 30
- Microsoft
  - dedicated configuration store, 137
  - operating system patches, 135
  - third-party authentication, 84
- Microsoft Active Directory, 105–106
- Microsoft Azure, 41
- Microsoft Internet Explorer, xiii
- Microsoft Windows, xiii
- minifying JavaScript files, 42–43, 115
- MODIFY statements (in SQL), 55
- MongoDB, 53
- monitoring, 44
- Mono project, 33
- Mosaic, xiii
- Mozilla Firefox, xiii
- Mozilla Foundation, 125
- multifactor authentication (MFA)
  - requiring, 89–90, 90*f*
  - third-party authentication, 84
- MX (mail exchange) records, 9, 86

## N

- National Center for Supercomputing Applications, xiii
- National Security Agency (NSA), 129
- .NET, 33
  - dependency checker, 136
  - securing XML parsers, 151
- Netflix
  - denial-of-service attacks, 163
  - technology blog, 168
- Netgear, 128
- Netscape, xiii, 95
- Nginx, 125–126, 132
- Node.js, 32, 133, 136, 151
- Node Package Manager (NPM), 136
- nonblind SQL injection attacks, 55
- NoSQL databases, 30, 53

- npm audit command, 136
- NSA (National Security Agency), 129
- nslookup command, 56–57

## O

- OAuth (open authentication) standard,
  - 84, 158
- obfuscating JavaScript files, 115
- object-relational mapping (ORM),
  - 54–55
- Offensive Security, 3, 136
- offloading static content, 167
- Okta, 84
- OneLogin, 84
- opaque IDs, 108, 111, 171
- open authentication (OAuth) standard,
  - 84, 158
- open directory listings, disabling, 137
- OpenID standard, 84
- open redirects, 153, 156–157, 172
- OpenSSL, 132
- openssl tool, 124–125
- Open Web Application Security Project (OWASP), 136
- operating system patches, 133–134
- OPTIONS requests (in HTTP), 11*t*
- Oracle VirtualBox, 3
- ORM (object-relational mapping),
  - 54–55
- os module (in Python), 62
- OWASP (Open Web Application Security Project), 136
- ownership-based access control, 106

## P

- PaaS (Platform as a Service), 41
- padding input data, 119
- parameterized statements, 52–53
- parent directories, 109, 112
- password-reset links, 87
- password-reset screens, 91
- passwords. *See also* authentication
  - commonly used, 84
  - cracking password lists, 89
  - hashing, 88–89, 119–120
  - requiring complex, 87–88
  - securely storing
    - hashes, 88–89
    - salting hashes, 89
  - securing resets, 87, 171

- password-strength-calculator library, 88
  - PATCH requests (in HTTP), 11
  - path separator character (/), 109
  - payloads, 141–142
  - penetration testing, 44, 160
  - Perl, 28
  - permissions, 171, 173
    - access control
      - access control lists, 105
      - aspects of, 104
      - audit trails, 107–108
      - common oversights, 108
      - defined, 104
      - implementing, 106–107
      - ownership-based access control, 106
      - role-based access control, 105–106
      - testing, 107
      - whitelists and blacklists, 105
    - directory traversal
      - absolute filepaths vs. relative filepaths, 108–109
      - anatomy of, 109–110
      - defined, 108
      - mitigation options, 110–112
      - privilege escalation, 104
  - Petrobras, 129
  - phishing, 154
  - PHP, xiv
    - command injection attacks, 56–58
    - file upload vulnerability attacks, 60–61
    - overview of, 32
    - vulnerability of, 32
  - ping floods, 164
  - ping of death attacks, 164
  - Platform as a Service (PaaS), 41
  - post-release activities
    - error reporting, 44–45
    - logging, 44–45
    - monitoring, 44
    - penetration testing, 44
  - POST requests (in HTTP), 11, 50
    - authentication, 83
    - cross-site request forgery attacks, 76–77
    - rendering pipeline, 20
    - R-U-Dead-Yet? attack, 165
  - pre-production environments. *See* test environments
  - primary keys (in SQL), 29
  - principle of least privilege, 55, 58, 152, 173
  - privilege escalation, 104
  - public-key certificates. *See* digital certificates
  - public-key cryptography, 119, 139
  - pull requests, 38
  - Puma, 125
  - Puppet, 42
  - PUT requests (in HTTP), 11, 76–77
  - Python
    - application servers, 125
    - command injection attacks, 56, 58
    - mitigating file upload vulnerability attacks, 62
    - overview of, 31–32
    - permissions, 106
    - securing XML parsers, 151
  - Python Software Foundation, 135
- Q**
- quality assurance (QA), 39
  - quality assurance environments. *See* test environments
- R**
- rainbow tables, 89
  - random number generation, 100
  - raw function (in Ruby), 68
  - RBAC (role-based access control), 105–106
  - React framework, 34, 72
  - reauthentication, requiring for sensitive actions, 79
  - reCAPTCHA widget, 92
  - Reddit, 84, 135, 166
  - Redis, 30–31, 53
  - reflected attacks, 165
  - reflected cross-site scripting attacks
    - defined, 70
    - escaping dynamic content, 71
  - regression testing, 135
  - regular expression (regex), 111
  - relational databases
    - consistent behavior, 29
    - data integrity constraints, 29
    - defined, 29
    - foreign keys, 29
    - primary keys, 29
    - SQL, 29–30, 50–51

- transactional behavior, 29
  - vulnerability of, 30
- relative filepaths, 109–110
- release process
  - automating, 41, 169
  - build process, 42–43
  - database migration scripts, 43
  - defined, 40
  - DevOps tools, 41–42
  - Infrastructure as a Service, 41
  - Platform as a Service, 41–42
  - pushing changes vs. releasing changes, 37
  - reliability of, 40
  - reproducibility of, 40–41
  - revertibility of, 41
- remote code execution attacks
  - anatomy of, 59
  - defined, 59
  - disabling code-execution during deserialization, 59–60
- rendering blink, 72
- rendering pipeline
  - Acid3 test, 18
  - defined, 15–16
  - Document Object Model, 16–17
  - styling rules, 16–18
- replay attacks, 139
- Representational State Transfer (REST), 76–77
- Rivest-Shamir-Adleman (RSA)
  - algorithm, 121
- role-based access control (RBAC), 105–106
- Rollbar, 44
- rolling back releases, 41
- root privilege, 104
- RSA (Rivest-Shamir-Adleman)
  - algorithm, 121
- Ruby and Ruby on Rails, 111, 125
  - ActiveRecord framework, 54–55
  - client-side error reporting, 115
  - client-side sessions, 96
  - command injection attacks, 56, 58
  - database migration scripts, 43
  - dependency checker, 136
  - overview of, 31
  - permissions, 106–107
  - securing XML parsers, 151
  - server-side sessions, 96–97
  - vulnerability of, 31, 59

- RubyGems package manager, 31
- R-U-Dead-Yet? (RUDY) attack, 165

## S

- S3 (Amazon Simple Storage Service), 62, 140
- SafeFrame standard, 142–143
- salting hashes, 171
- same-origin policy, 78
- SameSite cookie attribute, 78–79, 98–99
- SAML (Security Assertion Markup Language), 85
- sandboxing, 19, 141
- sanitizing client-side files, 116
- sanitizing file references, 111
- Sass, 43
- Scala, 32–33
- schemaless databases, 30
- Schneier, Bruce, 135
- `<script>` tags, 18–19
  - reflected cross-site scripting attacks, 70
  - stored cross-site scripting attacks, 66–67, 69
  - subresource integrity checks, 134
- SCSS, 43
- SDKs. *See* software development kits (SDKs)
- SDLC. *See* Software Development Life Cycle (SDLC)
- secure authentication system
  - banning disposable email accounts, 86, 87*f*
  - implementing and securing logout function, 90–91
  - preventing user enumeration
    - CAPTCHA, 91–92, 92*f*
    - error messages, 91
    - password-reset screen, 91
    - timing attacks, 91
  - requiring complex passwords, 87–88
  - requiring multifactor authentication, 89–90, 90*f*
  - requiring usernames, email addresses, or both, 85
  - securely storing passwords
    - hashes, 88–89
    - salting hashes, 89
  - securing password resets, 87
  - validating email addresses, 85–86

- Secure Hash Algorithm (SHA-256), 121
- Secure keyword, 98
- Security Assertion Markup Language (SAML), 85
- security certificates, 20
- security through obscurity, 108
- seeds, 100
- segregation of test and production environments, 39
- SELECT statements (in SQL), 50–51, 53, 55
- self-signed certificates, 124–125
- semicolon character (;), 52
- Sender Policy Framework (SPF), 155–156, 172
- SendGrid, 138, 156
- sequence numbers, 8
- serialization, 59
- serialization libraries, 59–60
- Server header (in HTTP responses), 114, 171
- server-side request forgery (SSRF) attacks, 150, 154, 159–160
- server-side sessions, 94–95
- session cookies, 95–97
  - cookie theft, 97–99
  - generic cookie parameters, 114
- session fixation, 99–100
- session hijacking
  - client-side sessions, 96–97
  - cookie theft, 97–99
  - defined, 93
  - opening sessions, 94
  - server-side sessions, 94–95
  - session fixation, 99–100
  - weak session IDs, 100
- session identifiers (session IDs)
  - session cookies, 95–99
  - taking advantage of weak, 100
  - TLS handshakes, 121
  - URL rewriting, 99–100
- session keys, 121–122
- session state, 94–96
- Set-Cookie header, 13, 20, 77–78, 90–91, 95–98, 171
- SHA-256 (Secure Hash Algorithm), 121
- SharkLasers, 87*f*
- Simple Mail Transport Protocol (SMTP), 154–155
  - defined, 10
  - Internet Protocol suite layers, 10*f*
  - single-page apps, 72
  - single quote character ('), 51–53
  - single sign-on (SSO), 84–85
  - Slowloris attack, 165
  - smoke testing. *See* post-release testing
  - SMTP. *See* Simple Mail Transfer Protocol (SMTP)
  - Snowden, Edward, 129
  - social media
    - database storage, 28, 66
    - likejacking, 158
    - logout function, 90
    - ownership-based access control, 106
    - permissions, 103–104, 106–107
    - posting links to external URLs, 158
    - SameSite attribute settings for cookies, 99
    - security advisories, 135
    - third-party authentication, 84
  - software development kits (SDKs)
    - avoiding server-side request forgery attacks, 160
    - defined, 31
  - Software Development Life Cycle (SDLC)
    - code writing
      - branching and merging code, 38
      - pushing changes to repository, 37
      - source control, 37
    - defined, 36
    - design and analysis, 36
    - post-release testing and
      - observation, 43–45
      - error reporting, 44–45
      - logging, 44–45
      - monitoring, 44
      - penetration testing, 44
    - pre-release testing
      - continuous integration servers, 39
      - coverage, 39
      - manual testing, 38
      - test environments, 39–40
      - unit testing, 39
    - release process, 40–43
      - build process, 42–43
      - database migration scripts, 43
      - DevOps tools, 41–42
      - Infrastructure as a Service, 41
      - Platform as a Service, 41

- source control (version control)
  - defined, 37
  - distributed vs. centralized, 37
  - pull requests, 38
- Space Jam* website, 24
- spam email and filters, 105, 154, 160
- spearphishing, 154
- SPF (Sender Policy Framework), 155–156, 172
- Splunk, 45
- spoofing, 50, 123, 153
- Spotify, 163
- SQL. *See* Structured Query Language (SQL)
- SQL injection attacks
  - anatomy of, 51–52
  - defined, 50
  - mitigation options
    - defense in depth, 55–56
    - object-relational mapping, 54–55
    - parameterized statements, 52–53
  - SQL, defined, 50–51
- SSO (single sign-on), 84–85
- SSRF (server-side request forgery)
  - attacks, 150, 154, 159–160
- Stack Overflow, 138
- staging environments. *See* test environments
- Stanford University, 7
- stateful connections, 13
- static resources
  - content delivery networks, 26
  - content management systems, 26, 27*f*
  - defined, 24
  - URL resolution, 24–25
- status codes, 12–13
- status messages, 12
- stored cross-site scripting attacks
  - content security policies, 69–70
  - escaping control characters, 67–69
  - example of, 66–67, 66*f*–67*f*
- Stripe, 138
- Structured Query Language (SQL)
  - databases, 29–30, 105
  - defined, 29, 50
  - typical statements, 50–51
- <style> tags, 17, 134
- styling rules and information (in CSS)
  - build process, 43
  - defined, 16
  - rendering pipeline, 16–18

- subresource integrity checks, 134
- symmetric encryption algorithms, 119
- SYN floods, 164
- system() function (in PHP), 58

## T

- TCP. *See* Transmission Control Protocol (TCP)
- templates, xiv
  - dynamic resources, 28
  - stored cross-site scripting attacks, 68–69
- test coverage, 39
- test environments (staging, pre-production, or quality assurance environments)
  - close resemblance to production environment, 39–40
  - defined, 39
  - hardening, 138
  - scrubbed data for, 40
  - segregation production environment and, 40
- testing
  - integration testing, 39
  - penetration testing, 44, 160
  - post-release, 43–45
  - pre-release, 38–40
  - regression testing, 135
  - unit testing, 39, 107, 170
- third-party authentication, 84
- third-party code
  - securing configuration, 136–138
    - disabling default credentials, 137
    - disabling open directory listings, 137
  - hardening test environments, 138
  - protecting configuration information, 137–138
  - securing administrative frontends, 138
- securing dependencies, 132–136, 171
  - deploying new versions quickly, 134
  - organizing dependencies, 132–134
  - staying alert to security issues, 135–136
  - timely upgrades, 136

- third-party code (*continued*)
  - securing services, 138–140
    - protecting API keys, 139
  - securing third-party content, 140
  - securing webhooks, 139
- third-party services risks, 140–143, 171
  - avoiding malware delivery, 141–142
  - malvertising, 141
  - reputable ad platforms, 142
  - reviewing and reporting suspicious ads, 143
  - SafeFrame standard, 142–143
  - tailoring ad preferences, 143
- time-to-live (TTL) variable, 9
- timing attacks, 91
- TinyLetter, 156
- TLS. *See* Transport Layer Security (TLS)
- Tornado, 125
- Tornado web server, 78
- Torvalds, Linus, 37
- TRACE requests (in HTTP), 11*t*
- traffic surges, 167, 173
- transactional behavior, 29
- transactional emails, 85, 156
- Transmission Control Protocol (TCP)
  - application layer protocols and, 9, 10*f*
  - checksums, 8
  - data packets, 8
  - denial-of-service attacks, 164
  - man-in-the-middle attacks, 118
  - origin of, 8
  - purpose of, 8
  - receipts, 8
  - rendering pipeline, 20
  - sequence numbers, 8
- transpiling, 34
- Transport Layer Security (TLS)
  - defined, 14, 118
  - handshakes, 14, 118, 120–122
    - cipher suites, 121
    - session initiation, 121–122
  - HTTP Secure, 14
  - message authentication codes, 120
- TTL (time-to-live) variable, 9
- Tumblr, 84
- Twitter, 76, 84, 135, 163
- TypeScript, 34, 42

## U

- UDP. *See* User Datagram Protocol
- UglifyJS utility, 115
- Unicorn, 125
- unintentional denial-of-service attacks, 166
- unit testing, 170
  - access control, 107
  - brittleness, 39
  - continuous integration servers, 39
  - defined, 39
  - test coverage, 39
- universal resource locators (URLs)
  - clean, 114
  - in HTTP requests, 10–11
  - open redirects, 157
  - relative vs. absolute, 157
  - static resources, 24–25, 25*f*
  - URI fragments, 71–72, 71*f*
  - URL rewriting, 99–100
  - URL-shortening services, 156
- UPDATE statements (in SQL), 50–51, 55
- URI fragments, 71–73, 71*f*
- URLs. *See* universal resource locators
- User-Agent header (in HTTP requests), 10–11, 50
- User Datagram Protocol (UDP), 8
  - denial-of-service attacks, 165
  - Internet Protocol suite layers, 10*f*
- users tables, 50–51

## V

- V8 JavaScript engine, 32
- validation tokens, 86–87
- verbs (methods) in HTTP, 10–11
- version control. *See* source control
- version numbers, 133
- vertical escalation, 104
- violation reports, 70
- VirtualBox, 3
- virtual containers, 3
- VPNFilter, 128
- Vue.js, 72

## W

- web components specification, 141
- webhooks, 139
- webmasters, xiv



- web page rendering
  - Acid3 test, 18
  - Document Object Model, 16–17
  - JavaScript, 16, 18–19
  - rendering pipeline, 15–19
  - styling rules, 16–18
- web programming languages
  - C#, 33
  - Java, 32–33
  - JavaScript, 32–34
  - Node.js, 32
  - PHP, 32
  - Python, 31–32
  - Ruby on Rails, 31
- web servers, xiv
  - defined, 23
  - directory traversal, 110
  - dynamic resources, 27–34
    - databases, 28–30
    - defined, 24
    - distributed caches, 30–31
    - templates, 28
    - web programming languages, 31–34
  - installing certificates
    - application servers vs., 125–126
    - configuring web server to use HTTPS, 126
  - remote code execution attacks, 59–60
  - static resources
    - content delivery networks, 26
    - content management systems, 26, 27*f*
    - defined, 24
    - URL resolution, 24–25

- web shells
  - file upload vulnerability attacks, 60–61
  - privilege escalation, 104
- where function (in ActiveRecord), 54
- whitelists, 105
- Wi-Fi hotspots, 128
- Wikipedia, 71–72
- wireless routers, 128
- wmap utility, 3, 4*f*
- WordPress, 26, 27*f*, 131, 138
- worms, 76
- WWW-Authenticate header, 82

## **X**

- XML. *See* Extensible Markup Language (XML)
- XML bombs, 148–149
- XML requests, 59
- XML Schema Definition (XSD) files, 147
- XMPP. *See* Extensible Messaging and Presence Protocol (XMPP)
- XSRF attacks. *See* cross-site request forgery (XSRF) attacks
- XSS attacks. *See* cross-site scripting (XSS) attacks

## **Y**

- YAML, 59, 146

## **Z**

- Zendesk, 26
- zero-day exploits, 2, 112
- zip bombs, 165