INDEX

Symbols and Numbers

; (semicolon), 110
-- (MySQL comment), 83, 84
<> (angle brackets), 53, 56
../ file path reference, 128
/ (forward slash), 99
| (pipe), 124
` (backtick), 122, 124
" (double quote), 56
' (single quote), 44–46, 56
# (hash), 44, 69
% (percent), 112
%00 (null byte), 99
%0A (line feed), 49
%0D (carriage return), 49
& (ampersand), 22–23, 110, 112
2FA (two-factor authentication), 183–184
32-bit processors, 133
64-bit processors, 133
127.0.0.1 (localhost), 102, 104–105
.docx file type, 113–114
!ELEMENT (XML), 110, 111–112
!ENTITY (XML), 110, 111–112
<img> tags, 32, 36–37, 63–65, 70, 171
<s> tag, 198

A

Abma, Jobert, 183–184, 198, 207–208
about:blank context, 57
Access-Control-Allow-Origin header, 34
access_denied parameter, 47
access_token (OAuth), 169–170
ACME customer information disclosure, 163–165
Ahrens, Julien, 101–104
alert function, 56, 65, 69–70
Algolia remote code execution bug, 125–127
Amass, 211
Amazon Simple Storage (S3)
- and bucket permissions, 181–183
- subdomain takeovers, 141–142

Amazon Web Services, 192
ampersand (&), 22–23, 110, 112
angle brackets (<>), 53, 56
AngularJS template engine
- injection examples, 73–74, 198–199
- Sandbox bypasses, 72–73

API *See* application programming interface (API)
apok (hacker), 186
application/json content-type, 33–34, 35
application logic and configuration vulnerabilities, 177–190
- GitLab two-factor authentication bug, 183–184
- HackerOne and S3 bucket permissions, 181–183
- HackerOne Hacktivity voting, 186–187
- HackerOne Signal manipulation, 180–181
- overview, 177–178, 189–190
- PornHub memcache installation, 188–189
- Shopify administrator privileges bypass, 179
- Twitter account protections, 180
- Yahoo! PHP info disclosure, 184–186

application programming interface (API), 7, 37–38, 90, 180, 197
application/x-www-form-urlencoded content-type, 32–34, 35
Aquatone, 194
A records, 140
arrays, 91–93
asset takeovers, 174–176. *See also* subdomain takeover vulnerabilities
Assis, Rodolfo, 69–70

authentication
HTTP requests, 50, 54, 150
misconfigurations, 173–174, 197
process, 30
Authmatrix plug-in, 160
`autofocus` attribute, 58
automation techniques, 185–186, 200
Autorize plug-in, 160
AWS metadata query bug, 100

B

Bacchus, Adam, 206
background jobs, 153–154, 156
backtick (`), 122, 124
Badoo full account takeover, 38–40
banking application illustrations
cross-site request forgeries, 29–30, 31–34
HTTP parameter pollution, 20–22
race conditions, 149–150
base64-encoded content, 9
bash, 120, 185–186
binary.com privilege escalation, 159–160
blacklisted characters, 52
blind SQLi, 84–87
blind SSRFs, 97–98
blind XSS attacks, 60, 198
Boolean attribute checks, 64, 86–87
Bounty Factory, 219
browsers
and cookies, 30–31
operations, 6–7
plug-ins for, 216
brute-forcing, 88–89, 195, 199, 211
Bryant, Matthew, 60, 223
Bucket Finder, 182, 214
Buerhaus, Brett, 99–100, 222
buffer overflow vulnerabilities, 130–133, 134–135
bug bounties, 2
platforms, 219–220
programs, 2, 90, 123, 188, 189, 203–204
Bugbounty JP, 219
Bugcrowd resources, 219, 222, 223
A Bug Hunter's Diary (Klein), 220
The Bug Hunters Methodology (Haddix), 220
bug reporting
after disclosures, 125
approach, 204–207
and hacker's reputation, 205–206
informative, 163–164
permission to test further, 76
proof of concept tips, 145
responses to, 16, 164–165
rewards appeals, 207–208
bugs previously reported, 125, 196
BuiltWith, 72, 213
Burp Suite, 40, 152, 158, 160, 195, 199–200, 210

C

Cable, Jack, 172
cache poisoning, 50
`call_user_func` (PHP), 121
Carettoni, Luca, 21, 22
carriage return line feed (CRLF)
CRLF injection vulnerabilities, 49–54
overview, 49–50, 54
Shopify response splitting, 51–52
Twitter response splitting, 52–54
Cascading Style Sheets (CSS), 6
C/C++ memory management, 129–133, 135
CDNs (content delivery networks), 144
censys.io website, 143, 214
certificate hashes tracking site, 143
Chan, Ron, 224
characters. *See also* sanitization of characters
blacklisted, 52–53
encoding, 42–45, 49, 88–90, 173–174
Charles (web proxy), 210
client-side HPP, 19, 22–23
client-side template injection (CSTI) vulnerabilities, 72–73, 73–74
clients
defined, 3
OAuth resource, 168–170
CNAME records, 140–146
Cobalt, 219
Coinbase comment injection, 42–43
comments in SQL queries, 83, 84, 92
companies
acquisition process exposures, 142
and bug bounty programs, 2, 204, 206–208
configuration vulnerabilities, 177–178
`CONNECT` method, 7–8

connection headers, 5
content attribute, 13, 45
content delivery networks (CDNs), 144
content discovery, 195
content spoofing, 41–42, 48
content-type headers, 6, 32–34, 35, 54
cookies
  and carriage return line feed injection, 50, 51–54
  in cross-site request forgeries, 32, 35–36
  in cross-site scripting, 56
  forgeries on, 126–127, 128
  operations and attributes, 30–31
  in subdomain takeovers, 140–141
CORS *See* cross-origin resource sharing (CORS)
Coursera, 218
CRLF characters *See* carriage return line feed (CRLF)
CRLF injection *See* carriage return line feed (CRLF), 49–54
cross-origin resource sharing (CORS), 34, 35, 38
cross-site request forgery (CSRF), 29–40
  Badoo full account takeover, 38–40
  defenses, 34–36
  Instacart, 37–38
  overview, 29–30, 40
  vs. server-side request forgeries, 95
  Shopify Twitter disconnect, 36–37
cross-site scripting (XSS) vulnerabilities. *See also* XSS Jigsaw blog; XSSHunter, 55–70
  and client-side template injections, 72
  Google image search, 65–66
  Google tag manager, 66–67
  overview, 55–58
  Shopify currency formatting, 62–63
  Shopify wholesale, 61–62
  types, 58–61
  United Airlines, 67–70
  Yahoo! Mail stored XSS, 63–65
crt.sh website, 143, 211
CSRF *See* cross-site request forgery (CSRF)
CSRF tokens, 33–35, 38–40, 45
CSTI *See* client-side template injection (CSTI) vulnerabilities
Cure53 Browser Security White Paper, 220
cURL requests, 124–125, 136
CVEs (disclosed security issues), 127
CyberChef, 44, 214

D

dangerouslySetInnerHTML function, 45, 72
databases, 150–151. *See also* SQL databases
db_query function (SQL), 92
De Ceukelaire, Inti, 44–46
DELETE method, 7–8
deserialization, 126–127
Detectify Labs, 112, 201, 223
dex2jar, 215
“did not respond”, 102
dig A command, 4
directory and file enumeration tools, 212
disclosed security issues (CVEs), 127
DNS *See* Domain Name System (DNS)
Document Object Module (DOM), 7, 13, 45
document parameters, 16, 56
document type definitions (DTDs), 108–110
domain cookie attribute, 30–31
Domain Name System (DNS), 3–4, 14, 97–98, 101–104, 141, 142
domain names, 3, 139–140
domain_name parameter, 14
DOM-based XSS, 59–60
Drupal SQLi, 90–93
DTDs (document type definitions), 108–110

E

Ebrietas (hacker), 144
EdOverflow (hacker), 146–147
email bug hunting examples, 74–76, 78–80, 87–90
Emek, Tanner, 154–155
encoded characters, 42–45, 49, 173–174, 197
error messages, 144
escapeshellcmd (PHP), 120–121
E-Sports Entertainment Association (ESEA) bug, 98–100
expandArguments function (SQL), 91–92
expires cookie attribute, 31
Exploit Database (DB), 195, 218

eXtensible Markup Language (XML), 110–117
- entities, 110
- overview, 107–110
- parsing and file types, 111–117

external HTTP requests, 96–97, 100–104, 104–105

EyeWitness, 127, 188, 212

F

Facebook
- and OAuth access token bug, 174–176
- ReactJS template engine, 72
- XXE with Microsoft Word bug, 112–114

Fastly, 144

Fehrenbach, Patrik, 66–67, 185–186, 222, 224

Fiddler (web proxy), 210

file and directory enumeration tools, 212

FileDescriptor (hacker), 46, 52–53, 59, 202, 224

file path expressions, 128

file types, 99, 114, 124–125, 197

file uploads, 122–123

filtered ports, 97

Firefox cookie bug, 52

firewall evasion, 50

flags on command line, 121

Flask Jinja2 template injection, 74, 123

Flurry password authentication, 172

forms
- hidden HTML, 33, 37
- as HTML injection, 42–43

forward slash (/), 99

FoxyProxy add-on, 216

Franjković, Josip, 152

`ftp_genlist()` function (PHP), 134–135

functionality mapping, 197–198

function execution, 121–122

fuzzing, 182

G

Gamal, Mahmoud, 159

`GET` requests
- in cross-site request forgeries, 31–32, 35, 40
- with open redirects, 12, 13
- operations, 7
- and server-side modifications, 36–37
- with SSRFs, 97

Ghostscript vulnerabilities, 202

Gill, Andy, 188–189, 224

GitHub, 126, 141, 178, 195

GitLab two-factor authentication bug, 183–184

Gitrob, 126, 195, 215

Gobuster, 195, 212

Google
- AngularJS template engine, 72–73, 73–76
- bug bounty program, 11
- dig tool, 101–104
- internal DNS SSRF, 100–104

Google bugs
- image search, 65–66
- tag manager, 66–67
- XXE vulnerability, 112

Google Chrome XSS Auditor, 59

Google dorking, 99, 100, 162, 195, 214

Google Gruyere, 218

Gowitness, 194, 212

H

The Hacker Blog, 223

Hacker101, 218

HackerOne bugs
- Hacktivity voting, 186–187
- interstitial redirect vulnerability, 13, 15–16
- invite multiple times, 150–151
- payments race condition, 153–154
- and S3 bucket permissions, 181–183
- Signal manipulation, 180–181
- social sharing buttons, 23–24
- unintended HTML inclusion, 44–47

HackerOne resources, 219, 221, 223

hacking blogs, 222–224

hacking techniques, 191–202
- efficiency suggestions, 200–202
- overview, 191–192, 202
- reconnaissance, 192–196
- testing, 196–200

Hacking: The Art of Exploitation (Erikson), 221

hacking tools, 214–215

Hack the Box, 218
Harewood, Philippe, 174–176, 201, 224
harry_mg (hacker), 142
Hasan, Mustafa, 67–70
hash (#), 44, 69
headers
  host and connection, 5
  injections, 50–52
HEAD method, 7–8
Heartbleed bug, 133–134
Heroku platform subdomain takeover example, 140–141
hidden HTML forms, 33, 37
Homakov, Egor, 178
Hopper, 216
Horst, Stefan, 90–91
host headers, 5
HPP *See* HTTP parameter pollution (HPP)
HTML *See* Hypertext Markup Language (HTML)
HTML injection vulnerabilities, 41–48
  Coinbase, 42–44
  examples, 42–47
  HackerOne, 44–47
  overview, 41–42, 48
  Within Security, 47–48
htmlspecialchars function, 23
HTTP *See* Hypertext Transfer Protocol (HTTP)
httponly cookies, 30–31, 50, 56, 185
HTTP parameter pollution (HPP), 19–27
  client-side, 22–23
  HackerOne social sharing buttons, 23–24
  overview, 19–21, 27
  server-side, 20–22
  Twitter unsubscribe notifications, 24–25
  Twitter Web Intents, 25–27
HTTP requests
  browser operations, 4–5
  external vs. internal traffic, 96
  methods, 7–8
  and race conditions, 150
  smuggling and hijacking, 50
  statelessness, 8–9, 30
HTTPScreenShot, 194, 213
HTTPS sites, 31
Hypertext Markup Language (HTML). *See also* HTML injection vulnerabilities
  character encoding, 42–43
  hidden forms, 33, 37
  rendering, 6
Hypertext Transfer Protocol (HTTP). *See also* HTTP parameter pollution (HPP); HTTP requests; HTTPScreenShot
  HTTPS sites, 31
  messages, 2
  response codes, 5, 6, 12
  response splitting, 50
  standards, 3

I

IDOR *See* insecure direct object reference (IDOR) vulnerabilities
id parameters, 121, 157–158
iFrames, 56, 69–70, 159–160
image file types, 124–125
ImageMagick software bugs, 123–125, 128, 202
<img> tags, 32, 36–37, 63–65, 70, 171
Inbound Parse Webhook, 146
IN clause (SQL), 91–92
innerHTML property, 54
input sanitization, 56, 61, 65, 120–121
insecure direct object reference (IDOR) vulnerabilities, 157–165
  ACME customer information disclosure, 163–165
  binary.com privilege escalation, 159–160
  Moneybird app creation, 160–161
  overview, 157–159, 165
  Twitter Mopub API token theft, 161–163
INSERT statements (SQL), 93
Instacart cross-site request forgery, 37–38
integer parameters, 25, 158, 161
internal DTD declarations, 109–110
internal server access, 96–97
Internet Archive Wayback Machine, 192
Internet Explorer
  CRLF injections, 52
  and Same Origin Policy, 57

Internet Protocol (IP). *See also* IP addresses, 3
interstitial web pages, 15–16
Intigriti, 220
introspection concept, 76
IP addresses
  ranges, 101–102, 104, 185–186, 193–194
  resolving, 3–4

J

Jamal, Mahmoud, 16, 38–40, 65–66
JavaScript
  and application logic vulnerabilities, 186–187
  for open redirects, 13, 16
  overview, 6–7
  and XSS payloads, 56–58, 61–62, 67–70
`javascript:alert(1)` payload, 65–66
JD-GUI, 216
Jinja2 template engine, 72, 74–76, 123

K

Kamkar, Samy, 55
Karlsson, Matthias, 196, 205
Kennedy, Justin, 96
kernel vulnerabilities, 122
Kettle, James, 73, 79, 224
Keybase invitation limit bug, 152
Kinugawa, Masato, 59
KnockPy, 141, 142, 211
krankopwnz (hacker), 51

L

Landry, Jasmin, 127–128
lcamtuf blog, 223
Legal Robot subdomain takeover, 144–145
Leitch, John, 135
libcurl read out of bounds bug, 136
Linux password storage, 111
Liquid Engine template engine, 62, 72
LiveOverflow, 222
local file disclosure, 127
localhost (127.0.0.1), 102, 104–105
local privilege escalation (LPE), 122
`Location` headers, 6, 12, 50, 54
`location` property, 13, 16
lock concept, 152, 155
logic problems *See* application logic and configuration vulnerabilities
login/logout CSRF, 60–61
logins. *See also* OAuth vulnerabilities
  authentication, 30
  phishing, 41–42
logouts and cookie expirations, 31
LPE (local privilege escalation), 122

M

mail exchanger (MX) records, 146
Markdown, 44, 46
mass assignment vulnerabilities, 178
Masscan, 213
`max-age` cookie attribute, 31
Meg tool, 195
memcache, 189
`memcpy()` method (C language), 135
memory management, 129–133, 136–137
memory vulnerabilities, 129–136
  buffer overflows, 130–133
  libcurl read out of bounds bug, 136
  overview, 129–130, 136–137
  PHP `ftp_genlist()` integer overflow, 134–135
  Python Hotshot module, 135
  read out of bounds, 133–134
metadata queries, 86–87, 100
Metasploit Framework exploits, 126–127
`<meta>` tags, 12–13, 45–46
Microsoft login tokens, 173–174
MIME sniffing, 6
mobile hacking, 200
mobile tools, 215–216
model, view, controller architecture (MVC), 77
Moneybird app creation, 160–161
Mozilla's bug tracker system, 221
MVC (model, view, controller architecture), 77
MX (mail exchanger) records, 146
Myspace Samy Worm, 55
MySQL, 82–83, 86–87

N

NahamSec blog, 223
`nc` command, 4
Netcat, 4, 125, 189
Nmap, 188, 193, 213

`nslookup` command, 188
null bytes, 99, 131
nVisium, 76–77

O

OAuth vulnerabilities, 167–176
  Facebook access tokens, 174–176
  Microsoft login tokens, 173–174
  overview, 167–170, 176
  stealing Slack tokens, 171
  Yahoo!-Flurry password authentication, 171–172
`onerror` attribute, 62, 64, 66, 69
`onfocus` attribute, 58
Online Hash Crack, 215
online training, 217–219
OOB (out-of-band) exfiltration, 98
open redirect vulnerabilities, 11–17
  HackerOne interstitial redirect, 13, 15–16
  overview, 11–13, 17
  Shopify login, 14–15
  Shopify theme install, 13–14
OpenSSL, 133–134
Open Web Application Security Project (OWASP), 11, 21, 112, 221
operating system vulnerabilities, 122
`OPTIONS` method, 7–8, 34, 35
Orange Tsai, 74–76, 87–90, 97, 123, 223
`Origin` header, 35
Ormandy, Tavis, 202
out-of-band (OOB) exfiltration, 98
OWASP *See* Open Web Application Security Project (OWASP)

P

packets, 2
Padelkar, Ashish, 181
page source view, 61
Paolo, Stefano di, 21, 22
Paraschoudis, Symeon, 136
password file exposure examples, 77, 79–80, 111–112, 121–122
paths, 5
payloads
  character encoding, 88–89, 198–199
  cross-site-scripting, 55–58, 61–62, 63, 65
PentesterLab, 218
percent (`%`), 112
"permission denied", 102
phishing attacks, 11, 42, 48
PHP
  arrays and functions, 91–93
  `call_user_func`, 121
  `escapeshellcmd`, 120–121
  file types, 122–123
  `ftp_genlist()` integer overflow, 134–135
  function execution, 121–122
  info disclosure bug, 184–186
  Smarty template engine, 72, 78–80
PHP Data Objects (PDO) extension, 90–93
`phpinfo` function, 185
`ping` command, 120–121
polyglots, 198
Polyvore website, 125
PornHub, 188–189
ports
  DNS lookup, 102
  and Same Origin Policy, 57
  scanning, 97, 104–105, 188–189, 193–194, 213
  uses of, 4
port scanning tools, 213
Portswigger Blog, 224
`POST` requests
  in cross-site request forgeries, 32–34, 37–38
  CSRF tokens in, 35, 40
  cURL options for, 124–125, 136
  operations, 8
  with SSRFs, 97
Prasad, Prakhar, 213
preflight `OPTIONS` calls, 8, 34
`prepareQuery` (SQL), 91
Prins, Michiel, 126–127, 209
Project Zero blog, 224
proxies *See* web proxies
Psyon.org IP address converter, 104
`PUT` method, 7–8
Pynnonen, Jouko, 64
Python Hotshot module vulnerability, 135
Python Jinja2 engine, 72

Q

quote characters, 56, 57. *See also* `"` (double quote); `'` (single quote)

R

race conditions, 149–156
  HackerOne invite multiple times, 150–151
  HackerOne payments, 153–154
  Keybase invitation limits, 152–153
  overview, 149–150, 156
  Shopify partners, 154–155
Rafaloff, Eric, 26–27
Rails *See* Ruby on Rails
Rails Secret Deserialization exploit, 126–127
Ramadan, Mohamed, 113–114
Rapid7
  on fuzzing, 182
  Rails Secret Deserialization, 127
RCE *See* remote code execution (RCE) vulnerabilities
React, 45, 186–187
ReactJS template engine, 72
read out of bounds vulnerabilities, 133–134, 136
reconnaissance, 192–196, 213–214
redirects
  OAuth, 168–170
  parameters, 12, 17
  responses to, 6, 12
  testing for, 96
`redirect_to` parameter, 12
redirect_uri (OAuth), 169, 171, 175
`Referer` header, 35
reflected XSS, 58–59
remote code execution (RCE) vulnerabilities, 119–128
  exploit on Algolia, 125–127
  overview, 119–123
  Polyvore and ImageMajick, 123–125
  through SSH, 127–128
`render` method, 77
Reni, Akhil, 161–163
Repeater tool, 158
Request for Comment (RFC) documents, 3
reserved characters, 42
resource owner (OAuth), 168–170
resource server (OAuth), 168–170
response_type (OAuth), 168–170
Rijal, Rohan, 145–147
rms (hacker), 179
root user access, 122, 127
Rosen, Frans, 145, 201
Ruby ERB template engine, 72, 77
Ruby on Rails
  configuration vulnerability, 178
  and cookie management, 126–127
  dynamic render bug, 76–77
  permissions validation, 179
  and SQLi countermeasures, 83–84
  URL pattern, 197

S

Sadeghipour, Ben, 100, 124–125, 128, 223
Same Origin Policy (SOP), 56–57
`samesite` cookie attribute, 35–36
Sandbox bypasses, 72–74, 75
sanitization of characters. *See also* unsanitized input exposures, 49, 54, 56, 198
scan.me subdomain takeover, 142
scopes (OAuth), 167–170
screenshotting, 194, 212–213
SecLists, 141, 195, 212
`secret_key_base` (Ruby on Rails), 126–127
`secure` cookie attribute, 31
Secure Socket Shell (SSH), 128
self XSS vulnerabilities, 60
semicolon (`;`), 110
SendGrid subdomain takeovers, 145–147
serialization, 126
server return messages, 102, 104–105
servers
  defined, 3
  responses, 5–6, 20–21
  staging and development, 188–189
server-side HPP, 19, 20–22
server-side request forgery (SSRF) vulnerabilities, 95–105
  ESEA bug and AWS Metadata query, 98–100
  Google internal DNS bug, 100–104
  internal port scanning, 104–105
  overview, 96–98, 113
server-side template injection (SSTI) vulnerabilities, 72, 74–75, 78–80
shell commands, 119–121, 122–123
`shell_exec` function, 120
Shodan, 214
Shopify bugs
  administrator privileges bypass, 179
  cross-site request forgeries, 36–37

currency formatting, 62–63
open redirect vulnerabilities, 13–15
partners race condition, 154–155
response splitting, 51–52
wholesale website, 61–62
Windsor subdomain takeover, 142–143
XSS, 61–63
Shopify Liquid Engine template, 62, 72
Silva, Reginaldo, 113
Slack OAuth token bug, 171
`sleep` command, 87, 90
Smarty template engine, 72, 78–80, 123
Snapchat Fastly subdomain takeover, 143–144
social engineering, 41–42, 48
software libraries as bug sites, 123, 125
SOP (Same Origin Policy), 56–57
Sopas, David, 115–117
source viewing, 61
Spelsberg, Max, 134
SQL databases
overview, 82–83
prepared statements, 83–84, 90–91
SQL injection (SQLi) attacks, 81–93
countermeasures, 83–84
Drupal SQLi, 90–93
overview, 81–83, 93
with SSRF responses, 98
Uber blind SQLi, 87–90
Yahoo! Sports blind SQLi, 84–87
sqlmap, 89, 215
SQL statements, 82–83
SSH (Secure Socket Shell), 128
SSL pinning, 200
SSL registration tracking sites, 143, 193
SSRF *See* server-side request forgery (SSRF) vulnerabilities
SSTI (server-side template injection) vulnerabilities, 72 , 74–75, 78–80
stack memory, 131–132
state (OAuth), 169
status codes, 5, 6, 13, 158
stored XSS, 59, 66–70, 100
subdomains
enumerating, 128, 188–189, 192–193, 211
overview, 139–140
subdomain takeover vulnerabilities, 139–147
Legal Robot takeover, 144–145
overview, 139, 141–141, 147, 189
scan.me pointing to Zendesk, 142
Shopify Windsor takeover, 142–143
Snapchat Fastly takeover, 143–144
Uber SendGrid mail takeover, 145–147
Ubiquiti CNAME example, 141–142
SubFinder, 192–193, 211
SUID (specified user ID), 122
Swinnen, Arne, 140–141
Synack, 220

T

The Tangled Web (Zalewski), 221
Tasci, Mert, 24–25
technology identification techniques, 196–197
template engines, defined, 71, 71–80
template injection vulnerabilities, 71–80
overview, 71–73, 80
Rails dynamic render, 76–80
Uber template injections, 73–76
testing methods, 196–200
`text/plain` content-type requests, 33
Thakkar, Jigar, 153–154
third party services exposures, 140, 142, 144–145, 146–147, 180, 197
tools list. *See also* hacking resources, 209–216
top-level domains, 139
`TRACE` method, 7–8
Transmission Control Protocol (TCP) connections, 4
Twitter bugs
account protections, 180
HTTP response splitting, 52–54
Mopub API token theft, 161–163
unsubscribe notification, 24–25
Web Intents, 25–27
Twitter security resource tweets, 221
two-factor authentication (2FA), 183–184

U

Uber bugs
AngularJS template injection, 73–74, 123
blind SQLi, 87–90
Jinja2 template injection, 74–76
Sendgrid mail takeover, 145–147

Ubiquiti subdomain takeover, 141–142
Udacity, 219
Ullger, Aaron, 180
Unicode characters, 52–53
Uniform Resource Identifier (URI), 7
Uniform Resource Locator (URL). *See also* HTTP parameter pollution (HPP); open redirect vulnerabilities
  defined, 7
  fragment, 69
  `name` parameters, 93
  parameter passing, 22–23, 47–48, 84–87
  parsing and decoding, 19–23, 173–174
  rendering, 57, 66, 98, 99
Unikrn bug, 78–80, 123
unintended actions, 2
universal unique identifiers (UUIDs), 158–159
unsanitized input exposures. *See also* cross-site scripting (XSS) vulnerabilities; remote code execution (RCE) vulnerabilities, 49
URI (Uniform Resource Identifier), 7
URL *See* Uniform Resource Locator (URL)
User Agent Switcher, 216
user id exploitation, 122
UUIDs (universal unique identifiers), 158–159

V

verification processes, 154–155
Vettorazi, Stefano, 84–87
view-source:URL, 61
virtual defacement, 41–42
virtual private server (VPS), 192
VPS (virtual private server), 192
vulnerabilities
  after code fixes, 46–47, 125
  defined, 2
vulnerability disclosure programs (VDPs). *See also* bug bounty programs, 2

W

Wappalyzer, 72, 78, 196, 216
Wayback Machine, 192
The Web Application Hacker's Handbook (Stuttard and Pinto), 198, 221
Web Development Tutorials YouTube channel, 222
web frameworks, 83–84
webhooks, 104–105, 146, 147
web page source view, 61
web proxies, 37, 158, 210–211
websites. *See also* domains
  browser access steps, 3–7
  new functionality exposures, 181, 186–187, 201
  redirection to malicious, 11, 12, 17
WeSecureApp (hacker), 36–37
Wfuzz, 212
What CMS, 214
white labeling, 146
white-listed assets, 34, 174–176
Whitton, Jack, 61, 173–174, 223
`whoami` command, 98
Wikiloc XXE, 115–117
wildcards
  and certificates, 143, 144
  and subdomains, 145, 147
`window.location` function, 13, 39–40
`window.onload` function, 39
Wireshark web proxy, 210
Within Security content spoofing, 47–48

X

XML *See* eXtensible Markup Language (XML)
XML External Entity (XXE) vulnerabilities, 107–117
  Facebook XXE with Microsoft Word, 112–114
  overview, 107, 111–112
  read access to Google bug, 112
  Wikiloc XXE, 115–117
XSS Auditors, 58–59
XSSHunter, 60, 198, 215
XSS Jigsaw blog, 224
XSS vulnerabilities *See* cross-site scripting (XSS) vulnerabilities
XXE *See* XML External Entity (XXE) vulnerabilities

Y

Yahoo! bugs
- Flurry password authentication, 172
- Mail, 63–65
- PHP information disclosure, 184–186
- Sports blind SQLi, 84–87

Yaworski, Peter, 104–105, 150–151, 160–161, 163–165, 181–183

ysoserial, 127, 215

Z

Zalewski, Michal, 223

ZAP Proxy, 37, 38, 211

Zendesk
- redirects, 15–16
- subdomain takeovers, 142

Zerocopter, 220

ZeroSec blog, 224

zseano (hacker), 143