INDEX

Symbols

& (ampersand), 208
* (asterisk), 203, 205
@ (at symbol), 54, 194
\ (backslash), 200
: (colon, for query aliasing), 50
{} (curly brackets), 47–48, 74–78, 176, 228
$ (dollar sign), 53
&& (double ampersand), 208
-- (double dash), 202
|| (double pipe), 208
" (double quotes), 197, 200
__ (double underscore), 78, 89–90, 94, 143, 150–151, 194
! (exclamation mark), 57
# (hash mark), 26
; (semicolon), 207–208
' (single quote), 197, 201
... (spread operator), 52–53
[] (square brackets), 7, 123
_ (underscore), 150

A

abstract syntax tree (AST), 8
adjacent server, 213
Agoo, 255–256
aliases, 50–52
  overloading, 116–119
Altair, 25–28
Amazon Web Services (AWS), 143–144
ampersand (&), 208
anti-CSRF tokens, 223, 228, 233–234, 240–241
Apache, 135
Apache Skywalking, 264–265
Apache Software Foundation, 95, 264
API Security Top 10, 190
Apollo Client, 8
AppImage extension, 25
application programming interface (API), 1
  consumers, 1–2
  endpoint paths, 15–16
  pagination, 122, 133
  scanners, 16, 164
arguments, 43, 48–50, 191–193
array-based query batching, 122–127, 253–255
ash_nz, 257
Assetnote, 36
asterisk (*), 203, 205
Atlassian, 2
at symbol (@), 54, 194
attack surface, 41–42
  matrix, 98
audit logs, 47
authentication and authorization
  flaws, 69, 163
  GraphQL Modules, 169
  GraphQL Shield, 169–170
  HTTP basic authentication, 165–166
  in-band vs. out-of-band, 164
  IP-based allow listing, 171
  OAuth, 167–169
  schema directives, 170–171
authorization grant, 167
autocorrection, 147
automatic persisted queries (APQ), 99, 134–135
awk, 141

B

backslash (\), 200
Base64, 166–167, 178

Bash
automating detection with cURL, 80
creating ID dictionaries, 184
for loop, 84, 109
formatting, 84
merging text files, 154
testing hosts, 109
BatchQL, 36, 126–127, 174, 232
batch requests, 99
blind SQL injection, 197
blind XSS, 213–214
Boolean-based SQL injection, 197
Boolean scalar type, 58
browser exploitation, 213
brute forcing, 173–177, 183–185
argument brute forcing, 183
CSRF token brute forcing, 234
password brute forcing, 173, 176
bug bounty, 66, 74, 248, 257, 260, 274
Burp Suite, 32–33, 197–200
BApp Store, 35
Byron, Lee, 148

C

caching, 134–135
canary GraphQL query, 85, 144, 172
central processing unit (CPU), 102, 128
cEWL, 154
circular relationships
fragments, 109, 255–256
introspection, 109, 251
queries, 102, 109, 118
claims, 168
Clairvoyance, 33–34, 152–154
classic SQL injection, 196–197
Classless Inter-Domain Routing (CIDR), 193–194
code review, 210–211
colon, for query aliasing (:), 50
comma-separated values (CSV), 40, 176, 184
Commix, 37–38, 208–210
Common Weakness Enumeration (CWE), 102
consolidator API layer, 8
cookies
authentication, 177
exfiltration, 211
session, 222, 240
cost analysis, 99, 128–130
cost analyzer, 128, 130
CrackQL, 40, 176–177, 183–184
credit-based rate limiting, 130
cross-site port attack (XSPA), 239
cross-site request forgery (CSRF)
GET-based cross-site request forgery, 228–231, 267–268
POST-based cross-site request forgery, 225–227
prevention, 232–234
cross-site scripting (XSS), 188, 211–214
blind, 213–214
DOM, 214
hunter, 214
reflected, 211–212
stored, 213–214
cross-site WebSocket hijacking (CSWSH), 46, 240–244
CS Money, 249
cURL, 24–25
curly brackets ({}), 47–48, 74–78, 176, 228

D

Damn Vulnerable GraphQL Application (DVGA), 29–31
changing modes, 93
installing, 28–29
rolling back, 215
security mechanisms, 93
data classification, 140
data field, 13, 74–75
datastore layer, 165
data types
enums, 58–59
inputs, 62–63
interfaces, 61–62
objects, 57–58
scalars
Boolean, 58
Float, 58
ID, 58
Int, 58
String, 58
unions, 60–61
Debian, 22

debug mode, 99, 157–158
declarative query language, 6, 14
decorate a field, 54
dee_see, 38
defense in depth, 234
denial of service (DoS), 69, 102, 248–256
dirbuster wordlist, 84
directives
  overloading, 119–121
  query-level
    @defer, 56
    @include, 54–56
    @skip, 54–56
    @stream, 56
  schema-level
    @auth, 170–171, 181–182
    @computed, 56
    @cost, 129
    @deprecated, 54–56
    @hasRole, 171
    @protect, 171
    @specifiedBy, 54, 56
disabled introspection, 93, 142–144, 148
Docker, 28–30
Document Object Model (DOM), 214
dollar sign ($), 53
double ampersand (&&), 208
double dash (--), 202
double pipe (||), 208
double quotes ("), 197, 200
double underscore (__), 78, 89–90, 94, 143, 150–151, 194
Doyensec, 34

E

edge, 5
edit-distance algorithm, 146–148
Elasticsearch, 187
enumeration, 182–183, 261–262
enum keyword, 58
enums, 58–59
error-based SQL injection, 196
errors field, 16, 75
escaping, 196, 201
exclamation mark (!), 57
Exploit Database, 21
Export SDL feature, 57
extensions field, 35, 130–131, 156–157
EyeWitness, 39, 82–83

F

Facebook, 2–3, 188
Fedora, 22
field
  defined, 4, 47
  duplication, 113–116, 132–133, 252–253
  stuffing, 149–154, 183
  suggestions, 48, 99, 145–148
File Transfer Protocol (FTP), 235
fingerprinting, server, 68, 94–98
Flask, 296
Float scalar type, 58
fragment keyword, 52
fragments, 52–53, 112
freddd, 252
fuzzing, 37, 141, 155, 177, 197, 208

G

gateway layer, 165
gateway proxies, 136–137
GitHub, 2, 29, 148
GitLab, 2, 251–252, 256, 259–262, 267–268
Google Chrome, 1, 233
graphical user interface (GUI), 17–20
GraphQL
  clients
    Altair, 25–28
    Apollo Client, 8
    cURL, 24–25
    GraphQL Explorer, 14, 17, 55, 81–83, 194–195
    GraphQL Playground, 14, 17, 55, 81–83, 194–195, 265–267
  data types
    custom objects, 57
    enums, 58–59
    inputs, 62–63
    interfaces, 61–62
    scalars, 58
    unions, 60–61

GraphQL *(continued)*
- endpoints
  - integrated development environment, 73
  - query, 73
- errors, 15–16, 75–76, 85, 95, 154–157, 172–173
- hacking tools
  - BatchQL, 36, 126–127, 174, 232
  - Burp Suite, 32–33, 197–200
  - Clairvoyance, 33–34, 152–154
  - Commix, 37–38, 208–210
  - CrackQL, 40, 176–177, 183–184
  - GraphQL Cop, 39, 127–128, 232
  - graphql-path-enum, 38, 182–183
  - GraphQL Threat Matrix, 66–67, 98, 132–135, 157
  - Graphw00f, 35–36
  - Netcat, 239, 241–244
  - SQLmap, 203–204
- implementations
  - Agoo, 255–256
  - Apollo Server, 72, 135, 142
  - Ariadne, 72, 88
  - AWS AppSync, 143–144
  - Graphene, 72, 74, 88, 98
  - graphql-go, 88
  - graphql-java, 72, 88
  - graphql-php, 68, 72, 88
  - graphql-ruby, 72, 88, 131, 135
  - Juniper, 72
  - Sangria, 72, 95–96, 160
  - Strawberry, 72
  - Tartiflette, 72
  - WPGraphQL, 253–255
  - Yoga, 72
- language
  - aliases, 50–52
  - arguments, 48–50, 191–193
  - directives, 54, 56
  - fragments, 52–53
  - operation name, 46, 177–178, 194–195
  - operation types, 43
  - query anatomy, 42
  - top-level fields, 43, 47
  - variables, 53–54
  - responses, 74–76
  - specification, 2–3, 111
  - spread operator (`...`), 52–53

graphql-directive-auth, 170
GraphQL Foundation, 2
GraphQL Modules, 169
graphql-scalars library, 50
GraphQL Shield, 169–170
GraphQL Threat Matrix, 66–67, 98, 132–135, 157
GraphQL Voyager, 92, 106
GraphQL Working Groups, 53
GTFOBins, 211

H

H2, 264
HackerOne, 248, 258–260, 262–263
handshake, 240–241
hash-based message authentication code (HMAC), 168
hash mark (#), 26
haxta4ok00, 258
headless browser, 39, 81
hypervisor, 22–23
HyperText Markup Language (HTML)
- `<a>` tag, 223, 228–230
- crafted forms, 222–223, 225
- errors, 124
- `<form>` tag, 223, 229
- heading, 218
- `<img>` tag, 223, 229–231
- injection, 230
- `<p>` tag, 218
- `<script>` tag, 212, 217, 218, 229
- `<title>` tag, 77, 214

HyperText Transfer Protocol (HTTP)
- basic authentication, 165–166
- client, 7, 24
- GET, 3, 25, 44, 77, 79, 160, 212, 223–224, 228–233, 242, 262–263, 267
- HEAD, 94–95
- headers, 27–28, 93, 171
- methods, 15, 157
- POST, 3, 15, 79–80, 160, 225–229
- query parameters, 79
- request methods, 15
- response codes, 15–16, 172

I

ID scalar type, 58
in-band authentication and authorization, 164
information disclosure, 69, 139–140, 261–262
information exposure through error messages, 154–155
injections, 69–70, 187–188, 190–195, 262–265
input/output (I/O), 128
InQL, 34–35, 107–108, 140–141
insecure direct object reference (IDOR), 177
interface keyword, 61
internal server, 189
introspection, 33, 63, 92, 99, 140, 142, 262
  query, 33, 142
  system types
    _Directive, 63
    _EnumValue, 63
    _Field, 63, 111
    _InputValue, 63
    _Schema, 63–64, 142–145
    _Type, 63–64, 143–144, 150–151
    _TypeKind, 63, 65
Int scalar type, 58

J

JavaScript, 2, 8, 50, 81, 211–219, 227
JavaScript Object Notation (JSON), 10
  GraphQL response, 74–76
    data field, 13, 74–75
    errors field, 16, 75
    extensions field, 130–131, 156–157
    location field, 155
    message field, 155
    path field, 156
Joaxcar, 256
Jobert, 262
Johnson, Donald B., 108
Johnson algorithm, 108
JSON Web Token (JWT), 167–169, 172, 178–180
Jumbo-WJB, 264

K

Kali, 23
key-value storage, 187

L

lab security precautions, 22
Lightweight Directory Access Protocol (LDAP), 235
limits
  alias and array-based batching, 132
  field duplication, 132–133
  query depth, 131–132
  returned records, 133
load balancer, 171
lowerCamelCase, 150
Lua language, 37
Lyon, Gordon (aka “Fyodor”), 37

M

Magento, 252–253
malicious input, 188
man-in-the-middle (MITM) attacks, 46
meta-fields, 78, 89, 143
microservices, 9
Microsoft SQL Server, 197
middleware, 132
Mozilla Firefox
  cookie modification, 87
  developer tools, 85
msdian7, 260
mutation keyword, 6
mutations, 6, 44
mvm, 250
MySQL, 8, 149, 187, 197, 203, 264

N

Neo4j, 187
Nginx, 95, 135, 160
Nikto, 81
Nmap Scripting Engine (NSE), 37, 76–77, 79
node package manager (npm), 267
nodes and edges, 4
non-null wrapping types, 57
normalization, 133
Nuri, 262

O

OAuth, 167–169
object limit overriding, 121–122
object types, 4
offensive security, 23
one-way link relationship, 5
on keyword, 52
OpenAPI, 66
OpenSearch, 264
operating system (OS) command injection, 205–210
operation names, 43, 46–47, 177–178, 194–195
out-of-band authentication and authorization, 164
over-fetching, 9
OWASP, 70, 189–190

P

payload, 168
penetration testing, 53, 101, 117–118
personally identifiable information (PII), 2, 69, 139, 149, 160–161, 196, 270
PhantomJS, 82
ports, 78
port scanning, 37
PortSwigger, 32, 44
PostgreSQL, 197, 203
programming languages
  Golang, 72, 88
  Java, 72, 88, 265
  JavaScript, 2, 8, 50, 81, 211–219, 227
  PHP, 35, 68, 72, 88
  Python, 34, 72–74, 80, 118, 124, 206, 244
  Ruby, 72, 88, 131, 255, 263
  Rust, 72–73
  Scala, 72–73, 95, 160
  TypeScript, 8, 72
protected resource, 167
public server, 189

Q

query
  allow list, 133
  batching
    alias-based, 50–52, 173–175
    array-based, 122–127
  cost analysis, 99, 128–131
  depth limit, 99, 131–132
  height, 133
  keyword, 6–7
  parsers, 3, 7–8
  tracing, 250–251

R

Rapid7, 261
React, 81
reconnaissance, 71
recursive queries, 102, 109, 125
Redis, 187
reference implementation, 2, 148
reflected cross-site scripting (XSS), 211–212
regular expressions
  online evaluator, 250
  ReDoS, 50, 249–250
Relay, 8
request for comments (RFC), 3, 15–16, 165, 168, 178
request forgery, 211
resolver functions, 3, 8, 210–211
resource owner, 167
REST APIs, 2, 7–12, 15, 48, 66
return all, 192
reverse proxies, 171
Rivest-Shamir-Adleman (RSA), 168
role-based access controls, 164
root type, 6
Ruby, 72, 88, 131, 255, 263

S

SameSite, 232–233
sanitization, 47, 188, 295
scalar keyword, 58
scalar types, 4, 50, 58, 89
  Boolean, 58
  Float, 58
  ID, 58
  Int, 58
  String, 58
schema, 3–4
  directives, 170–171, 181–182
  federation, 9
  stitching, 9
schema definition language (SDL), 4, 92–93

security developments, 148
segmentation fault, 112
selection set, 47–48, 74–78
Selenium, 82
semi-blind SSRF, 235–236
semicolon (`;`), 207–208
serialization, 13, 58
Server Message Block (SMB), 235
server-side request forgery (SSRF), 207, 234–239
  prevention, 240
SHA-256, 134
Shopify, 257–258
shorthand query syntax, 47–48
SIGINT signal, 116, 120
signature, 167–169, 179–180
single quote (`'`), 197, 201
snake_case, 150
snapshots, 22
SOAP, 70, 249
SpectaQL, 93
spread operator (`...`), 52–53
SQL injection (SQLi), 264–265
  automating, 203–205
  blind, 197
  Boolean-based, 197
  classic, 196–197
  error-based, 196
  testing for, 197–203
  time-based, 197, 264
  union-based, 196
SQLite, 202
square brackets (`[]`), 7, 123
stack traces, 158–160
Star Wars API (SWAPI), 183
Stasinopoulos, Anastasios, 37
state-changing action, 223
static analysis, 128
stored cross-site scripting, 213–214
stored payload, 213–214
`String` scalar type, 58
strongly typed, 8, 193
structured query language (SQL)
  engine, 201
  operator, 201
  query, 201, 264
  table, 201
Stupin, Nikita, 34
`subscription` keyword, 6
subscriptions, 6, 44, 231, 240–244
Swagger, 66

T

tab-separated values (TSV), 140
Tarjan algorithm, 108
TiDB, 264
time-based SQL injection, 197, 264
timeouts, 135
Transport Layer Security (TLS), 46, 166
Truncer, Chris, 39
Tsaturov, Ilya, 34
Twitter, 188
two-way link relationships, 5
`type` keyword, 4
TypeScript, 8, 72
type stuffing, 150–152
type validation, 8

U

Ubuntu, 22
Uncontrolled Resource Consumption, 102
under-fetching, 9
underscore (_), 150
unified gateway, 9
union-based SQL injection, 196
`union` keyword, 60
unions, 60–61
unordered arguments, 49

V

validation, 3, 66
variables, 53–54
Vazarkar, Rohan, 39
VirtualBox, 22–23
virtualization, 22
vulnerability assessment, 16, 94

W

web application firewalls (WAFs), 17, 136, 143–144, 174
WebSocket, 44–46, 200, 217, 231, 240–244, 262
wordlist, 74, 81–84, 152–154, 176, 184

WordPress, 253
WPGraphQL, 253–255

X

X-Forwarded-For, 171
X-Host, 171
X-Originating-IP, 171
X-Real-IP, 171
XSS. *See* cross-site scripting
XSS Hunter, 214

Z

zerodivisi0n, 262