

2ND  
EDITION

# WEBBOTS, SPIDERS, AND SCREEN SCRAPERS

A GUIDE TO DEVELOPING INTERNET AGENTS  
WITH PHP/CURL

MICHAEL SCHRENK



# INDEX

## Symbols & Numbers

- & (ampersand), in GET method, 67
- \$address array, 156–157
- \$content\_type variable, 158
- \$data\_array, 168
  - for LIB\_http library functions, 35
- \$FETCH\_DELAY, 175, 180
- \$filter\_array, 130, 135–136
- \$\_GET array, 76
- \$link\_array elements, 111
- \$page\_base variable, 106
- \$\_POST array, 76
- \$result array, FILE element, 51–52
- \$status\_code\_array, 114–115
- . (period), as POP3 end-of-message indicator, 147
- ? (question mark), in GET method, 67
- 404 Not Found error, 287, 338

## A

- abstractions, of program interface, 81
- access log file, 29
  - error logging in, 266
  - and webbot detection, 266–267
- access rights, verifying, 20
- action attribute
  - of form, 65,
  - for form analyzer, 71
- action of person, simulating, 123
- \$address array, 156–157
- agent name
  - default for, LIB\_http, 31
  - defining for PHP/CURL session, 330

- log record of, 268
- spoofing, 29, 280, 311, 316, 329
- aggregating information by relevance, 16
- aggregation webbots, 92, 129–137
  - CDATA, 135
  - choosing data sources, 130
  - downloading and parsing script, 134
  - and filtering, 135–137
  - RSS feeds, 131–133
  - writing, 133–135
- Alexa web-monitoring service, 305
- “all rights reserved” notice, 320
- Amazon Web Services, SOAP interfaces, 305
- ampersand (&), in GET method, 67
- anchor tags. *See* links
- Andreessen, Marc, 1
- anonymity
  - as a process, 283
  - in commercial email, 155
- anti-pokerbot software, 18
- Anti-Spam Law, Virginia, 324
- Apache
  - cookies, 202
  - headers, 33, 190
  - installing PHP on, 30
  - log files, 266–267
  - web server, 6
- Application Program Interfaces (APIs)
  - Amazon, 131
  - eBay, 131
  - Google, 127
  - Google Maps, 131

- archive\_links() function, 178
  - ARPANET, 139
  - array
    - assigning parsed data to, 98
    - elements, form data as, 68
    - of <img> tags, src attribute from, 61
    - parsing
      - data set into, 41–42
      - table into, 96
  - attributes, parsing values, 42–43
  - audience, for Internet, 2
  - authentication, 190–208
    - basic, 199–202
      - curl\_setopt() function options for, 332
      - by PHP/CURL, 28
      - test pages, 201
    - of buyer by procurement
      - webbot, 186–187
    - default response to request, 28
    - for deterring webbots, 314
    - digest, 201–202
    - and encryption, 202
    - example scripts and practice pages, 199
    - FTP, 140
    - with query string sessions, 205–207
    - session, 202–205
    - of snipers, 189, 213
    - strengthening by combining techniques, 198–199
    - types, 198
  - automating tasks, 19
- B**
- bandwidth
    - consumption, 187, 225
    - hijacking, 104, 291
    - stealing, 30
  - base64-encoding, 84
  - basic authentication, 199–202
    - curl\_setopt() function options for, 332
    - by PHP/CURL, 28
    - test pages, 199
  - batch file, for webbot, 216
  - Bcc: address field, 156–157
  - Beck & Tysver legal website, 319
  - Bidder’s Edge spiders, 323
  - bids, timing placement of, 188, 191
  - Bina, Eric, 1
  - binary-safe download routine, 103–104
  - biometrics, 198–199
  - Bing, spiders used by, 173
  - blobs, storing images as, 83–84
  - blogs
    - aggregation of, 131
    - laws concerning, 324, 325
    - searching for spelling errors, 137
  - botnet management, 255–260
    - assigning tasks, 258–260
    - communication methods, 255
    - determining tasks, 257
    - performing tasks, 260
    - polling the botnet server, 256–257
    - task checkout, 258
    - uploading botnet data, 260
  - broken links, webbot detecting, 109–116
  - browser buffering, 27
  - browser-like webbots, 230
  - browser macros, 227–247
    - adding functionality to, 240–247
    - browser-like webbots, 230
    - commands, 235
    - creating your first, 231–233
      - initialization, 233
      - recording, 232
    - defined, 230
    - dynamic macros, 241–245
      - integrating data with, 242–245
      - scripts that create, 241
    - hacking, 239–247
    - installing, 230–231
    - iMacros Scripting Engine, reasons not to use, 240–241
    - launching automatically, 245–246
      - in Linux, 246
      - in Windows, 245

- necessity for, 237
- overcoming barriers with, 230
- reasons to use, 227–229
- running, 237
- suggested standard initialization
  - of, 235–237
- browsers, 1–2
  - emulating, 75. *See also* browser macros
  - executing webbots in, 26–27
  - inspiration from limitations of, 15–18
  - problem with, 2
  - search engine treatment vs. treatment of webbot, 126
  - tabbed browsing, 129
- business leaders, webbot benefits for, 11–12
- buy-it-now auction purchases, 189

## C

- CamelCase, 78
- CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), 314–315, 325
- Cascading Style Sheets (CSS), 17, 230
  - impact of removing HTML tags, 89
- case
  - for naming, 111
  - sensitivity, `stristr()` function vs. `strstr()` function, 137
- Cc: address field, 156–157
- CDATA tags, 134–135
- certificates, 195
- Children’s Online Privacy Protection Act (COPPA), 154
- ciphers, 193, 195
- client-server technology, 2
- client URL Request Library (cURL), 6, 21, 23
- clipping service, online, 20, 137
- clocks, synchronization for sniper, 189

- code
  - in book, 4–5
  - libraries available online, 5
- collusion webbots, 17
- command shell
  - executing webbots in, 26
  - leveraging operating system with, 254
  - and spider scripts, 181
- comma-separated value (CSV) files
  - `file()` function for downloading, 27
  - iMacros file format, 236
- Common Object Request Broker Architecture (CORBA), 305
- communication, on incompatible systems, 21–22
- competitive advantage, 9–13, 64, 74, 191, 265, 286, 310
- Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA), 314–315, 325
- compressing data, 86–88
- computers. *See also* server
  - distributing tasks across multiple, 254
- constructive hacking, 11
- Content-Type line
  - for email message, 148
  - in HTTP header, 34
- `$content_type` variable, 142
- converting website into function, 163–170
- COOKIE\_FILE, 212–214
- cookies
  - about, 209–211
  - adapting to management changes, 294
  - for authentication, 202–205
  - defaults for, 31
  - deleting, 212, 294
  - for deterring webbots, 313
  - expiration dates for, 212–213
  - and forms, 70

- cookies, *continued*
    - managing multiple users', 213–214
    - persistence with, 212
    - PHP/CURL to read and write, 29
    - purging temporary, 212–213
    - restrictions, with proxies, 206
    - viewing, 210–211
    - and webbot design, 211
  - COPPA (Children's Online Privacy Protection Act), 154
  - copyright issues, 85, 319–322
    - "all rights reserved" notice, 320
    - and facts, 321
    - fair use laws, 322
    - registration, 320
  - CORBA (Common Object Request Broker Architecture), 305
  - crawlers. *See* spiders
  - cron command, 215
  - cryptography, 193
  - CSS (Cascading Style Sheets), 17, 230
    - impact of removing HTML tags, 89
  - CSV (comma-separated value) files,
    - file() function for downloading, 27
    - iMacros file format, 236
  - cURL (client URL Request Library), 6, 21, 23
  - curl\_error() function, 334–335
  - curl\_exec() function, 334
  - curl\_getInfo() function, 334
  - curl\_init() function, 328
  - curl\_setopt() function, 69, 104, 328–333, 334
    - case sensitivity, 333
    - CURLOPT\_COOKIEFILE option, 205, 214, 331
    - CURLOPT\_COOKIEJAR option, 205, 214, 331
    - CURLOPT\_FOLLOWLOCATION option, 329
    - CURLOPT\_HEADER option, 330
    - CURLOPT\_HTTPHEADER option, 331
    - CURLOPT\_MAXREDIRS option, 288, 329
    - CURLOPT\_NOBODY option, 330
    - CURLOPT\_PORT option, 333
    - CURLOPT\_POSTFIELDS option, 332
    - CURLOPT\_POST option, 332
    - CURLOPT\_REFERER option, 329
    - CURLOPT\_RETURNTRANSFER option, 329
    - CURLOPT\_SSL\_VERIFYHOST option, 195
    - CURLOPT\_SSL\_VERIFYPEER option, 195, 332
    - CURLOPT\_TIMEOUT option, 294–295
    - CURLOPT\_UNRESTRICTED\_AUTH option, 332
    - CURLOPT\_URL option, 329
    - CURLOPT\_USERAGENT option, 330
    - CURLOPT\_USERPWD option, 332
    - CURLOPT\_VERBOSE option, 333
    - executing, 333–334
  - custom logs, and webbot detection, 268
- ## D
- daily scheduling of webbots, 217
  - data
    - fields in forms, 65, 66
    - networks, access and abuse, 323
    - set, parsing into array, 41
    - sources, choosing for aggregation webbot, 130
  - \$data\_array, 168
    - for LIB\_http library functions, 35
  - database
    - for saving links, 181–182
    - storing images in, 83–84
    - storing text in, 80–83
  - data management, 77–90
    - organizing data, 77–85
      - naming conventions, 77–78
      - storing images in database, 83–84
      - storing text in database, 80–83
      - structured files, 79–80

- reducing size, 85–90
  - data compression, 86–88
  - removing formatting, 88–89
  - storing references to image files, 85
- thumbnailing images, 89–90
- data-only interfaces, 301–307
  - lightweight data exchange, 302–305
- REST (Representational State Transfer) 306–307
- SOAP (Simple Object Access Protocol), 170, 305–306
- XML (eXtensible Markup Language), 131, 301–302
- <data> tags, for insertion parse, 123–124
- dates, in filenames, 79–80
- DCOM (Distributed Component Object Model), 305
- decode\_zipcode() function, 165
- deep linking, 291
- default file, for web page, 24
- delays, inserting between page fetches, 270
- DELE command (POP3), 149
- deleting
  - cookies, 212
  - HTML formatting, 88–89
  - unwanted text, 43–44
  - white space, 89
- delimiters
  - parsing text between, 40
  - splitting string at, 39
- deployment of webbots. *See* scaling
- denial-of-service (DoS) attacks, preventing, 180, 252–253, 330
- DES (Digital Encryption Standard), 195
- describe\_zipcode() function, 167–169
- developers, webbot benefits for, 9–11
- difficult websites, scraping, 227–247
- digest authentication, 201–202
- digital certificate, 194–196

- Digital Encryption Standard (DES), 195
- directories, 79
  - script for creating, 104
- disclaimer, 6
- disk swapping, 181
- Distributed Component Object Model (DCOM), 305
- <div> tags, parsing data into array, 95
- DOS (denial-of-service) attacks, preventing, 180, 252–253, 330
- download\_binary\_file() function, 103–104
- download\_images\_for\_page() function, 105
- downloading
  - with FTP, 139–143
  - with LIB\_http, 23–35
  - with link-verification webbot, 109–110
  - linked page, 113
  - with PHP built-in functions, 25–27
  - with PHP/CURL, 27–35
  - web pages, 23–35
- download\_parse\_rss() function, 133, 134

## E

- eBay, 19, 130, 151, 188, 237, 306, 310, 323
  - snipers and, 187
- Electronic Frontier Foundation (EFF), 325
- email
  - guidelines, 154
  - headers, 148
  - keeping legitimate out of spam filter, 158–159
  - for notification
    - of FTP transmission failure, 140–141
    - of webbot action, 161
  - placing account information in script, 150

- email, *continued*
  - reading with webbots, 145–152
  - sending, 153–161
    - HTML-formatted, 159–160
    - with mail() function, 154–155
    - notifications with webbots, 157–158
    - with PHP, 154–155
  - undeliverable as alert to invalid address, 160–161
  - as webbot trigger, 223
- email-controlled webbots, 151–152
- encryption, 193–196
  - authentication and, 208
  - certificate, 195–196
  - for deterring webbots, 312
  - webbots using, 194
- end-of-message indicator (POP3), 147
- environments, 250–252
  - many-to-many, 251
  - many-to-one, 252
  - one-to-many, 250
  - one-to-one, 251
- error
  - handlers, 295–296
  - information
    - from http\_get() function, 32
    - from http\_get\_withheader() function, 32
  - logs, and webbot detection, 267–268
- eval() function, 303
- event triggers, 70
- exclude\_link() function, 179–180
- exclusion list, for spiders, 180
- executing webbots
  - in browsers, 26–27
  - in command shell, 26
- exe\_sql() function, 82–83
- expiration dates, for cookies, 203, 209–210
- eXtensible Markup Language (XML), 301–302
  - assigning tasks, 258, 260
  - overhead, 302
  - for RSS feeds, 131

## F

- facts, and copyright, 321
- fair use laws, 322
- fault-tolerant webbots, 285–296
  - cookie management
    - changes, 294
  - form changes, 292–293
  - network outages and congestion, 294–295
  - page content changes, 291–295
  - URL changes, 286–291
    - page redirection, 288–290
    - and referer values accuracy, 290–291
    - requests for nonexistent pages, 292
- \$FETCH\_DELAY, 175, 180
- fgets() function, 25, 27
- file() function, downloading files with, 27
- file handle, 25, 27
- filesystem, geographically structured, 80
- File Transfer Protocol (FTP)
  - server, connecting to, 141
  - webbots, 139–143
- \$filter\_array, 130, 135–136
- filtering
  - by aggregation webbot, 135–137
  - information by relevance, 16
- Flash
  - barrier to effective webscraping, 229
  - for deterring webbots, 314
  - for website navigation, problems caused by, 229
- fopen() function, 25
- format of names, 78–79
- formatted\_mail() function, 156
- form data variables, 66
- forms
  - adapting to changes in, 292–293
  - analyzing, 71–74, 165
  - avoiding errors, 75–76
  - and cookies, 70
  - emulation, 64
  - legal issues and, 64

- handlers, 65–66
- input tags, 66
- interfaces, reverse engineering, 64–65
- main parts, 65
- source code
  - displaying, 166
  - saving, 166
- submission, 63–76, 167
  - data fields in forms, 66
  - event triggers, 70
  - form handlers, 65–66
  - GET method, 67–68
  - PHP/CURL for, 28
  - POST method, 68
  - unpredictability, 70
- <form> tag, action attribute, 66
- fputs() function, 89, 107, 149
- From: address field, 156
- FTP (File Transfer Protocol)
  - server, connecting to, 141
  - webbots, 139–143
- ftp\_cdup() function, 142
- ftp\_chdir() function, 142
- ftp\_delete() function, 142
- ftp\_get() function, 142
- ftp\_mkdir() function, 142
- ftp\_put() function, 142
- ftp\_rawlist() function, 142
- ftp\_rename() function, 142
- ftp\_rmdir() function, 142
- fully resolved URLs, 212
- functions. *See also individual function names*
  - converting website into, 163–170
    - describe\_zipcode() function, 167–169
  - interface definition, 168
  - submitting form, 168
  - target page analysis, 165–167

## G

- garbage collection, by PHP, 335
- geographically structured
  - filesystem, 80
- \$\_GET array, 76
- get\_attribute() function, 42–43, 61, 107

- get\_base\_page\_address() function, 106
- get\_domain() function, 178–179
- get\_http() function, 95
- GET method, 61
  - and errors, 267
  - http\_get() function for down-  
loading with, 31–32
  - vs. POST method, 68
- Google
  - bombing, 298
  - developer API, 127
  - spiders used by, 173
- GoogleRankings.com, 118
- graphics. *See* images

## H

- hacking
  - constructive, 11
  - iMacros, 239–247
  - webbot activity misinter-  
preted as, 266
- handle for file, 25
- handshake process, 195
- hard drives, compressing files on, 87–88
- hardware requirements, 5–6
- harvest, separating from
  - payload, 181
- harvest\_links() function, 177–178
- hash, 157–158
- haystack, 44
- header tags, and search engine optimization, 299
- headers
  - in email, 147–148
  - redirection, 113, 288
- <head> tag, detecting redirection, 288–290, 312
- Hello World! web page, 25
- hijacking bandwidth, 104, 291
- holidays, scheduling
  - webbots on, 270
- Hormel Foods Corporation, 153*n*
- hotel room prices, aggregating and filtering data, 16
- href attribute
  - extracting value, 112
  - of link tag, parsing, 42–43



- HTML (Hypertext Markup Language)
    - for formatting email, 159–160
    - parsing
      - content of reoccurring tags, 41–42
      - poorly written web pages, 38
      - text between tags, 40
    - removing formatting, 88–89
  - htmlspecialchars() function, 313
  - HTMLTidy (Tidy), 38, 46
  - HTTP
    - header, 31–32
      - exchanging cookies in, 202
      - and security, 68
    - protocol, 25
    - port for, 256
    - status codes, 133–134, 337–338
  - HTTP codes, 337–338
    - from http\_get\_withheader() function, 33
  - http\_get\_form() function, 35
  - http\_get\_form\_withheader() function, 35
  - http\_get() function, 31–32, 35
  - http\_get\_withheader() function, 32–33, 35
  - http\_header() function, 35
  - http\_post\_form() function, 35, 168
  - http\_post\_withheader() function, 35
  - http() routine, 31
  - HTTPS protocol, 194
  - human patterns, webbot simulation of, 269–272
  - Hypertext Markup Language. *See* HTML (Hypertext Markup Language)
- I**
- iMacros. *See* browser macros
  - image-capturing webbots, 101–108
    - binary-safe download routine, 103–104
    - directory structure, 104–105
    - execution, 103
    - main script, 105–107
  - image-processing loop, 107
  - images
    - borrowing from other sites, 104
    - storing in database, 83–84
    - thumbnailing, 89–90
  - <img> tags
    - alt attribute, 300
    - parsing from downloaded web page, 106–107
    - src attribute from array, parsing, 43
  - incompatible systems, communication on, 21–22, 151
  - index file, for web page, 24
  - indexing web pages, by search engine spider, 300
  - infinite loops, preventing, 330
  - information, aggregating and filtering by relevance, 16
  - initialization
    - download\_images\_for\_page() function, 102–103
    - link-verification webbot, 109–110
    - search-ranking script, 121–123
  - input tags in forms, 66
  - insert() function, 81–82
  - insertion parse, 123–125
  - installing
    - HTMLTidy, 28
    - iMacros, 230–231
    - PHP/CURL, 30
  - intellectual property law, 318–319
    - protecting, 19–20
  - interfaces, data-only, 301–307
  - Internet
    - access to, 6
    - audience for, 2
    - customizing for business, 12
    - law, 324–325
  - Internet Explorer, setting webbot name to, 75
  - Internet Protocol (IP) addresses, 275–276
  - intranet, 6
  - IP (Internet Protocol) addresses, 275–276

## J

- Java applets, for deterring webbots, 314
- JavaScript
  - for data manipulation, 70
  - deleting, 43–44
  - for deterring webbots, 313
  - as event trigger, 70
  - impact of removing
    - HTML tags, 89
  - impact on spider indexing, 180
  - redirection with, 290

## K

- Kelly v. Arriba Soft*, 322, 324
- keywords
  - in meta tags, 299
  - spamming, 299

## L

- landmark, 96
  - for end of data, 96
  - to identify table, 241
  - for table heading row, 96
  - using least likely to change, 291
- legal issues. *See also* copyright issues
  - for email, 154
  - in form emulation, 64
  - Internet, 324–325
  - website policies and, 316
- legitimate mail, keeping out of
  - spam filters, 154
- LIB\_download\_images library, 102
- LIB\_http\_codes library, 114, 337–338
- LIB\_http library
  - default conditions for, 31
  - downloading with, 28–35
  - file for storing cookies, 205
  - for form analysis emulation, 71–74
  - for form emulation, 67–69
  - source code, 34
    - defaults, 30, 35
    - functions, 31–34, 35

- LIB\_mail library, 156–157
- LIB\_mysql library, 80, 81
  - exe\_sql() function, 81–82
  - insert() function, 81–82
  - update() function, 82
- LIB\_parse library, 37–46
- LIB\_pop3 library, 149
- LIB\_resolve\_addresses library, 109
- LIB\_rss library, 133
- LIB\_simple\_spider library, 176–180
- LIB\_thumbnail library, 89–90
- lightweight data exchange, 302–307
- \$link\_array elements, 111
- link-verification webbots, 109–115
  - advanced options, 115
  - displaying page status, 114
  - downloading linked page, 113
  - flowchart, 110
  - generating fully resolved URLs, 112–113
  - initialization and downloading target, 109–110
  - parsing links, 111
  - running, 114–115
  - setting page base, 111–112
  - verification loop, 111–112
- links
  - broken, using webbot to detect, 109–115
  - href attribute of tag, parsing, 42–43
  - impact of removing HTML tags, 88–89
  - parsing, 111
  - relative, page base for, 106
  - saving in database, 181–182
  - well-defined, and search engine ranking, 289
- Linux, scheduling in, 215
- LIST command (POP3), 147
- Location: line, in HTTP header, 288
- log files
  - software for monitoring, 268–269
  - webbot detection with, 266–269
- logging in, to POP3 mail server, 146
- login criteria, 198

## M

- Mac OS X, scheduling in, 215
- macros, browser. *See* browser macros
- mail() function, 154–155
- maximum penetration level for spider, 174
- Message Digest Algorithm (MD5), 195, 202
- meta tags, 41–42, 127, 299
- MIME type, 34, 148, 159, 307, 331
- mkdir() function, 104–105
- mkpath() function, 102, 105
- monthly scheduling of webbots, 217, 219, 220
- Mosaic, 1
- MSN, spidering Google, 127
- MySQL, 6, 80, 81–84

## N

- naming
  - conventions, 78–79
  - data fields, 66
  - webbots, 75
- National Oceanic and Atmospheric Association (NOAA), 163
- needle, 44
- network
  - socket, 25
  - adapting to outages and congestion, 286
- Next button, simulating person clicking, 123
- NOAA (National Oceanic and Atmospheric Association), 163
- nofollow option, for robots meta tag, 312
- noindex option, for robots meta tag, 312
- non-ASCII content, and search engine spiders, 301
- nonexistent web pages
  - avoiding requests for, 286–288
  - containing forms, 292
  - timeouts to deal with, 331
- null string, replacing text with, 45

## O

- obfuscation, 193, 313
- obsolete web pages, risk of targeting, 287
- online
  - auctions, automating bidding in, 19, 63
  - clipping service, 20–21
  - purchases, automating, 185–192
- opening tags, for function parameter, 41
- open proxies, 279–280
- opensocket() function, 149
- optimizing website performance, 17
- organic placements in search results, 118–119
- organizing data, 77–85
  - naming conventions, 77–78
  - storing images in database, 83–85
  - storing text in database, 80–83
  - structured files, 79–80
- outgoing header message, from PHP/CURL session, 331
- overhead, in XML file, 302

## P

- package-tracking information, 145
- packet sniffer, 208*n*, 237
- page base
  - defining, 106
  - setting, 110–111
- \$page\_base variable, 106
- page redirection, 288–290
  - CURLOPT\_FOLLOWLOCATION option for, 329
  - for deterring webbots, 313
- page signature, 157
- paid placements in search results, 118–119
- parse\_array() function, 41–42, 52, 61–62, 95, 106, 111, 125
- parse tolerance, 291
- parsing, 37–62
  - attribute values, 42–43
  - data set into array, 41–42

- image tags from downloaded web page, 106
- with LIB\_parse, 39–44
- links, 111
- poorly written HTML, 46
- position vs. relative, 291–292
- with regular expressions, 49–62
- src attribute, from array of <img> tags, 41
- standard routines for, 38
- text between delimiters, 40
- unformatted text, 45
- passwords, 198
  - for deterring webbots, 314
- pattern matching, with regular expressions, 50
  - alpha, 53
  - alternate matches, 54
  - grouping, 55
  - numbers, 53
  - character sets, 53
  - ranges, 55
  - wildcards, 54
- payload for spider, 175, 181
  - separating from harvest, 182
- pay-per-click advertising, 325
- PEAR (PHP Extension and Application Repository), 305
- penetration level for spider, 174
- period (.), as POP3 end-of-message indicator, 147
- periodicity of webbots, 217, 225
- permanent cookies, 209–210
- persistence with cookies, 209
- phishing attack, 154
- phone numbers, parsing with regular expressions, 55–59
- PHP, 4–5, 6
  - configuring to send email, 154–155
  - downloading
    - with built-in functions, 25–27
    - with scripts, 23
  - and FTP, 142
  - functions, 44–46
    - for compressing data, 86–87
  - and SSL, 194
  - version 5 support for SOAP, 305
  - website, 6
- PHP/CURL, 28
  - and certificates, 195
  - and cookies, 202
  - downloading with, 28–30
  - encryption and, 194
  - for following header redirections, 288, 329
  - installing, 30
  - sessions
    - closing, 335
    - creating minimal, 327–328
    - initiating, 328
    - retrieving information about, 334
    - setting options, 328–333
    - viewing errors, 334–335
- PHP Extension and Application Repository (PEAR), 305
- php.ini* file, editing to show mail server location, 154–155
- plotting Wi-Fi networks, 21
- pokerbots, 17–18
- POP3 protocol (Post Office Protocol 3), 146–152
  - authentication failure, 146
  - executing commands with webbots, 149–151
- port
  - for HTTP and HTTPS protocols, 194
  - for POP3 server, 146
- position parsing, avoiding, 291
- \$\_POST array, 76
- POST method, 68–69
  - and errors, 267
- Post Office Protocol 3 (POP3), 146–152
  - authentication failure, 146
  - executing commands with webbots, 149–151
- preg\_match\_all() function, 51
- preg\_match() function, 51
- preg\_replace() function, 50
- preg\_split() function, 52

- price-monitoring webbots, 93–100
    - parsing script, 96–99
    - target, 94
  - procurement bot, 185–192
    - purchase criteria, 188
    - purchase triggers, 187
    - theory, 186–191
  - project ideas, 15–22
    - automating tasks, 19
    - communicating on incompatible systems, 21–22
    - consolidating industry news articles, 18–19
    - intellectual property protection, 19–20
    - online clipping service, 20
    - plotting Wi-Fi networks, 21
    - pokerbots, 17–18
    - tracking web technologies, 21
    - verifying access rights, 20
    - WebSiteOptimization.com, 17
  - projects
    - aggregation webbots, 129–137
    - converting website into function, 163–170
    - FTP webbots, 139–143
    - image-capturing webbots, 101–108
    - link-verification webbots, 109–115
    - price-monitoring webbots, 93–100
    - search-ranking webbots, 117–127
    - sending email with webbots, 152–161
    - reading email with webbots, 145–152
  - proxies, 273–284
    - commercial, 282
    - cookie restrictions with, 206
    - creating a service, 283–284
    - defined, 273–274
    - listing services, 280
    - open, 277–280
      - anonymous, 280
      - dark side of, 280
      - spoofing, 280
      - transparent, 280
    - reasons developers use, 274–277
      - anonymity, 274–276
      - relocation, 277
  - Tor, 281–282
    - configuration for
      - PHP/CURL, 282
      - disadvantages of, 282
    - using, 277
      - in a browser, 278
      - with PHP/CURL, 278
  - public, capitalizing on inexperience with webbots, 12
  - purchase
    - criteria, for procurement bot, 186
    - triggers, for procurement bot, 187
- ## Q
- query string sessions, authentication with, 205–207
  - question mark (?), in GET method, 67
  - QUIT command (POP3), 149
- ## R
- random delay, 123
  - ranking web pages, by search engine spider, 298
  - reading mail from POP3 server, 145–152
  - realm, 200
  - Real Simple Syndication (RSS) feed, 130, 131–132
  - redirection, 288–290
    - CURLOPT\_FOLLOWLOCATION option for, 329
    - for deterring webbots, 313
    - with PHP/CURL, 29
  - references to image files, storing, 85
  - referer
    - management, with
      - PHP/CURL, 30
    - variable, 32, 73, 104, 329

- regular expressions, 39, 49–62
    - advanced parsing with, 49–62
    - avoiding, 39, 47
    - disadvantages of, 60–61
      - complicating code, 61
      - confusing choices, 61
      - difficulty debugging, 61
      - lack of context, 60
    - functions, 50–52
      - `preg_match()`, 51
      - `preg_match_all()`, 51
      - `preg_replace()`, 50
      - `preg_split()`, 52
    - resemblance to PHP
      - built-in, 52
    - pattern matching with, 50
      - alpha, 53
      - alternate matches, 54
      - character sets, 53
      - grouping, 55
      - numbers, 53
      - ranges, 55
      - wildcards, 54
    - parsing phone numbers with, 55–59
    - speed of, vs. PHP built-in functions, 62
    - types of, 50
      - PCRE, 50
      - POSIX, 50
    - when to use, 60
  - relational database, 77
  - relative links, page base for, 106, 110, 111, 115
  - relay host, 155
  - relevance, aggregating and filtering information by, 15
  - Remote Procedure Call (RPC), 305
  - remote server, using PHP/CURL to execute webbot on, 216
  - `remove()` function, 43–44
  - replacing portion of string, 45
  - Reply-to: address field, 157
  - Representational State Transfer (REST), 306–307
  - `resolve_address()` function, 113
  - resources, distributing, 169–170
  - respect, 318–319
  - REST (Representational State Transfer), 306–307
  - `$result` array, FILE element, 51–52
  - RETR command (POP3), 147–148
  - `return_between()` function, 40, 61, 134
  - Return-path: address field, 157
  - reverse engineering form interfaces, 64–65
  - robot exclusion file, 311
  - robots meta tag, 312
  - robots.txt* file, 311–312
  - root
    - directory, creating for imported file structure, 106
    - domain, parsing from target URL, 178–179
  - RPC (Remote Procedure Call), 305
  - RSET command (POP3), 149
  - RSS (Real Simple Syndication) feed, 130, 131–132
- ## S
- sale item, verifying availability, 187
  - saving
    - links in database, 181–182
    - source code for form, 165
  - scaling, 249–262. *See also* botnet management
    - causing DoS attacks, 252–253
    - environments 250–252
      - many-to-many, 251
      - many-to-one, 252
      - one-to-many, 250
      - one-to-one, 251
    - multiple instances, creating, 253–254
      - distributing tasks, 254
      - forking, 253–254
      - leveraging the operating system, 254
  - scheduling, 215–225
    - adding variety to, 225
    - complex, 218–219
    - disabling, 296
    - for distributed spider, 183

- scheduling, *continued*
  - and stealth, 270
  - webbots to run daily, 217–218
  - webbots to run monthly, 219
  - Windows 7 Task Scheduler, 220, 223
  - Windows XP Task Scheduler, 216–219
- scraping, difficult websites, 227–237
- scripts, 3, 4–5
  - writing in small steps, 46
- search engine
  - optimization, 118, 252, 297–300
  - spiders, 173
    - design techniques hindering, 315–316
    - indexing web pages with, 298
  - Terms of Service agreement, 126, 310
- search-ranking webbots, 117–127
  - fetching search results, 123
  - how they work, 120–121
  - initializing variables, 121–122
  - parsing search results, 123–126
  - running, 120
  - search results page description, 118–119
  - starting loop, 122
  - what they do, 120
- search results page, parts of, 118–119
- search term, in URL, 122
- Secure Sockets Layer (SSL), 193–194
  - CURLOPT\_SSL\_VERIFYHOST
    - option for, 195
  - CURLOPT\_SSL\_VERIFYPEER
    - option for, 195
  - sites, downloading images
    - from, 103–104
- seed URL, 174
- sending email, 153–161
- server
  - avoiding undue load on
    - target, 324
  - error log, form errors in, 75–76
  - obtaining clock value, 189–190
  - remote, using PHP/CURL to
    - execute webbot on, 216
- session
  - authentication, 202–207
  - ID, forms with, 66
  - with proxies, 278
  - value, dynamically assigned, 167–168
- set\_time\_limit() function, 175, 295
- Short Message Service (SMS), 161, 341–344
- Simple Object Access Protocol (SOAP), 170, 305–306
- simulating action of person, 269–270
- single points of failure,
  - avoiding, 225
- size reduction, 85–90
  - data compression, 86–88
  - removing formatting, 88–89
  - storing references to image files, 85
- SMS (Short Message Service), 161, 341–344
- snipers, 185–192
  - authentication, 189
  - clock synchronization, 189–190
  - testing, 191
- SOAP (Simple Object Access Protocol), 170, 305–306
- socket management, with PHP/CURL, 30
- software
  - for monitoring logs, 268–269
  - requirements for, 6
- source code
  - configuration area of
    - LIB\_mysql, 81
  - for form
    - displaying, 165
    - saving, 166
- spam, 153–154, 255, 298
  - filters, 154
  - keeping legitimate mail out of, 158–159
  - keywords, 299
  - law, 324
- spam indexing, 298
- special characters, 122, 313

- spiders, 173–183
    - adding payload, 181
    - distributing tasks across multiple computers, 182
    - examples, 175–176
    - experimenting with, 180–181
    - how they work, 174
    - LIB\_simple\_spider library, 176–180
      - archive\_links() function, 178
      - exclude\_link() function, 179–180
      - get\_domain() function, 178–179
      - harvest\_links() function, 177–178
    - maximum penetration level for, 174
    - options for treating unwanted, 316
    - potential ideas for, 173–174
    - regulating page requests of, 183
    - saving links in database, 181–182
    - of search engines, 126–127
    - setting traps for, 315–316
    - what to do with unwanted, 316
  - split\_string() function, 39
  - splitting string, at delimiter, 39
  - SQL (Structured Query Language), 80
  - src attribute, from array of <img> tags, parsing, 43
  - SSL (Secure Sockets Layer), 193–194
    - CURLOPT\_SSL\_VERIFYHOST option for, 195
    - CURLOPT\_SSL\_VERIFYPEER option for, 195
    - sites, downloading images from, 103–104
  - \$status\_code\_array, 114–115
  - status codes, 337–339
    - HTTP, 337–338
    - NNTP, 339
  - status of request, from
    - http\_get\_withheader() function, 32
    - status messages, quantity created in file transfer, 333
  - stealth, 265–272
    - reasons for, 265–266
    - and scheduling, 270
    - simulating human patterns in order to achieve, 269–270
  - Stenberg, Daniel, 28
  - strings
    - detecting within strings, 44–45
    - measuring similarity of, 46
    - replacing portion of, 45
    - splitting at delimiter, 37
  - strip\_cdata\_tags() function, 39
  - strip\_tags() function, 61, 88
  - stristr() function, 44–45
  - strops() function, 124
  - str\_replace() function, 45
  - strstr() function, 45
  - structured files, 79–80
  - Structured Query Language (SQL), 80
  - submit button, 66
  - substr() function, 52, 124
  - synchronization, 21–22, of clocks for snipers, 189
- ## T
- tables
    - parsing data in, 96
    - using landmarks to identify, 291–292
  - tags. *See individual tag names*
  - targets, 3–4
    - validation in
      - download\_images\_for\_page() function, 102–103, 105
  - target URL, defining for PHP/CURL session, 329
  - tasks, automating, 19
  - Task Scheduler (Windows 7), 220–223
  - Task Scheduler (Windows XP), 216–219
    - complex scheduling, 218–219



- Telnet, 2, 28
  - for executing POP3 commands, 146–148
- temporary cookies, 209–210
  - purging, 212–213
- Terms of Service agreements, 126, 310
  - for search engines, 118
- text
  - embedding in other media, 314–315
  - messaging, 161, 341–344
  - parsing unformatted, 45
  - removing unwanted, 43–44
  - storing in database, 80–83
- thumbnailing images, 89–90
- Tidy (HTMLTidy), 38, 46
- time
  - required for downloading linked pages, 114
  - running webbot during busy, 270
- timeout
  - curl\_setopt() function for, 294, 331
  - default for, 31
    - and spiders, 175
  - in PHP, changing, 295
  - for PHP/CURL, 294, 331
- timestamp, Unix, 167
- <title> tag, and spiders, 298
- TLS (Transport Layer Security), 194
- Tor, 281–282
  - configuration for PHP/CURL, 282
  - disadvantages of, 282
- tracking web technologies, 21–22
- TrackRates.com, 16
- transactional websites, 192
- transfer protocols, PHP/CURL support for, 28
- Transport Layer Security (TLS), 194
- trespass-to-chattels law, 241, 253, 271, 322–324
- triggers, non-calendar-based, 223–224
- trim() function, 61
- Tysver, Daniel A., 319

## U

- undeliverable mail, using to prune access lists, 152
- unformatted text, parsing, 45
- unique keywords, 299
- Unix
  - scheduling in, 215
  - timestamp, 190
- unsubscribe options, for email 154
- unwanted text, deleting, 43–44
- update() function, of LIB\_mysql, 82
- updating website, frequency for deterring webbots, 314
- uploading files, with FTP, 141
- urlencode() function, 112
- URLs
  - adapting to changes, 286–291
    - page redirection, 288–290
    - referrer values' accuracy, 290–291
  - requests for nonexistent pages, 286–291
  - defining target for PHP/CURL session, 329
  - fully resolved, 112–113
- US Copyright Office, 320, 321
- usernames, 199

## V

- validation point, for downloaded web page, 287–288
- variables, passing to webbots, 304–306
- verification loop, 110–111
- Virginia, Anti-Spam Law, 324
- virtual private networks (VPNs), 198
- virtual property, laws governing, 324
- VPNs (virtual private networks), 198

## W

- W3C (World Wide Web Consortium), HTTP codes, 113*n*
- weather forecasts, 163
- web agents, selectively allowing access to specific, 311–312

- webbot\_error\_handler() function, 295
- WEBBOT\_NAME constant, 75
- webbots (web robots), 2
  - benefits of, 9–10
    - for business leaders, 11–12
    - for developers, 10–11
  - cookies and design of, 212–214
  - countermeasures for, 309–316
    - with cookies, encryption, JavaScript, and redirection, 313
  - embedding text in other media, 314–315
  - obfuscation, 313
  - reasons for, 309
  - robots meta tag, 312
  - robots.txt file, 311–312
    - allowing selective access to specific agents, 312–313
    - Terms of Service agreements, 126, 310
  - creating first script, 25
  - daily scheduling of, 217–218
  - executing
    - in browsers, 26–27
    - in command shell, 26
  - fault-tolerant, 286–296
  - growth in use, 10
  - monthly scheduling of, 219
  - periodicity of, 217, 225
  - preparing to run as scheduled tasks, 216
  - preventing negative consequences of, 317–325. *See also* copyright issues
  - project ideas, 15–22
  - for reading email, 145–152
    - and executing POP3 commands, 149–151
    - and POP3 protocol, 146–149
  - reasons for stealth, 265–272
  - script, creating first, 25
  - for sending email, 153–161
  - setting traps, 315–316
  - simulating human patterns, 269–272
  - spreading burden of running complex, 169–170
  - testing, 191
  - and trespass-to-chattels law, 241, 253, 271, 322–324
  - weekend scheduling of, 270
- web pages
  - accessibility to webbots, 297–300
  - adapting to content changes, 291–292
  - avoiding requests for nonexistent, 286–288
  - displaying status of, 113–114
  - notification of change in, 157–160
  - parsing image tags from downloaded, 106
  - poorly written HTML within, 38
  - ranking by search engine spider, 298
  - status of request for, 337–338
  - validation point for, 287
- web services, 305
  - designing custom lightweight, 302–305
- websites
  - for book, 4
  - converting into functions, 163–170
  - limiting access to, 197–208
  - optimizing performance of, 17
  - transactional, 192
- web spiders. *See* spiders
- web technologies, tracking, 21
- web walkers. *See* spiders
- WebSiteOptimization.com, 17
- weekends, scheduling webbots not to run on, 270
- well-defined links, for search engine optimization, 298
- white space, deleting, 45, 89
- Wi-Fi networks, plotting, 21
- Windows Task Scheduler
  - Windows 7, 220–223
  - Windows XP, 216–219
    - complex scheduling, 218–219
- wireless subscriber, mail server, 342
- World Wide Web, 1

World Wide Web Consortium  
(W3C), HTTP codes, 113*n*  
wrapper function, using  
    PHP/CURL within, 30

## **X**

XML (eXtensible Markup Lan-  
guage), 301–302  
    assigning tasks, 258, 260  
    overhead, 302  
    for RSS feeds, 131  
<xmp> and </xmp> tags, 26  
    displaying parses within, 47

## **Y**

Yahoo!, spiders used by, 173

## **Z**

ZIP codes  
    database for, 170  
    web page for decoding, 164–166