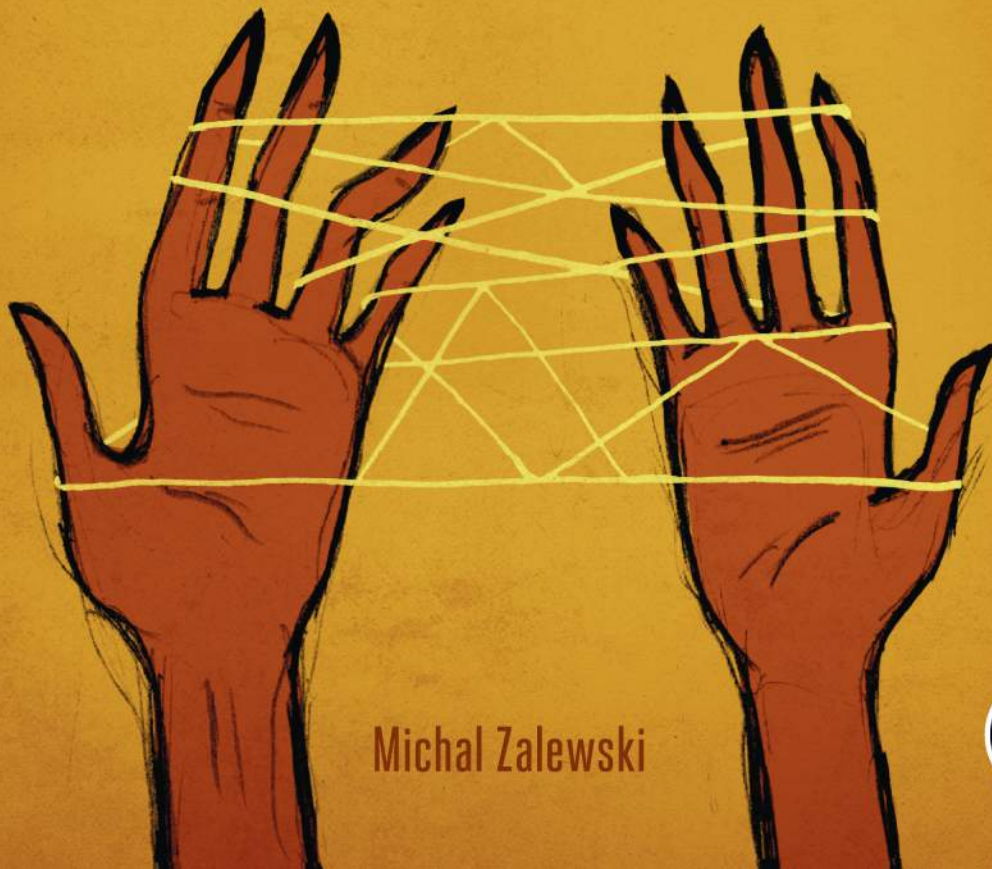


the Tangled Web

*A Guide to Securing Modern
Web Applications*



Michał Zalewski



INDEX

Symbols & Numbers

& (ampersand), in HTML, 71
<> (angle brackets)
 browser interpretation, 74–75
 in HTML, 71
<![CDATA[...]]> blocks, 72, 78, 250
<!DOCTYPE> directive, 71
<!ENTITY> directive, 76
<!-- and -->, for HTML comments, 72
<% ... %> blocks, Internet Explorer
 and, 75
@ directives, in CSS, 89–90
\ (backslashes) in URLs, browser accep-
 tance of, 29
` (backticks), as quote characters, 74, 111
!- directives, 76
// fixed string, in URLs, 25
% (percent sign), for character
 encoding, 31
. (period), hostnames with, and cookie-
 setting algorithms, 159
?-directives, 76
<?xml-stylesheet href=... ?> directive, 88
; (semicolon), as delimiter
 in HTTP headers, 48–49
 in URLs, 29
200–299 status codes, 54
300–399 status codes, 55
400–499 status codes, 55–56
500–599 status codes, 56

A

 tag (HTML), 79
 target parameter, 174–175
about:blank document, origin inheritance,
 165, 166–167
about:config (Firefox), navigation risks, 188
absolute URLs, vs. relative, 25
Accept-Language request header, 43

Accept request header, 43
Access-Control-Allow-Origin header,
 237–238, 240
acrobat: scheme, 36
action parameter, for <*form*> tag, 80
ActionScript, 132–134
Active Server Pages, 75
ActiveX, 129, 136–137
address bars, 220
 and EV SSL, 65
 hiding, 221
 manipulation, 256–257
Adobe Flash, 119, 130, 132–134
 and cross-domain HTTP headers, 147n
 file handling without *Content-Type*, 199
 HTML parser offered by plug-in, 133
 policy file spoofing risks, 156–157
 security rules, 154–157
Adobe Reader, 130
Adobe Shockwave Player, 132
ADS (Alternate Data Stream) Zone
 Identifier, 231
advertisements, new window for, 217
Akamai Download Manager, 137
Allow-forms keyword, for *sandbox*
 parameter, 246
AllowFullScreen parameter, for Flash, 155
AllowNetworking parameter, for Flash, 155
Allow-same-origin keyword, for *sandbox*
 parameter, 246
AllowScriptAccess parameter, for Flash, 154
Allow-scripts keyword, for *sandbox*
 parameter, 246
Allow-top-navigation keyword, for *sandbox*
 parameter, 246
Alternate Data Stream (ADS) Zone
 Identifier, 231
ambient authority, 60, 60n
ampersand (&), in HTML, 71
anchor element (HTML), specifying
 name of, 28

- angle brackets (< >)
 - browser interpretation, 74–75
 - in HTML, 71
- anonymity, scripts and, 249
- anonymous requests, in CORS, 239
- anonymous windows, 175
- antimalware, 236n
- Apache
 - and *Host* headers, 47
 - `PATH_INFO`, 201
- APNG file format, 83
- Apple QuickTime, 119, 130, 132
- Apple Safari. *See* Safari (Apple)
- `<applet>` tag (HTML), 83, 128, 135, 183
- `application/binary`, 212
- `application/javascript` document type, 118
- `application/json` document type, 118, 202
- `application/mathml+xml` document type, 119
- `application/octet-stream` document type, 200–201, 212
- `application/x-www-form-urlencoded`, 81
- Arce, Ivan, 2n
- Arya, Abhishek, 209
- asynchronous *XMLHttpRequest*, 146
- Atom, 123
- `<audio>` tag (HTML), 84, 119
- authentication, in HTTP, 62–63
- authorization, vs. authentication, 62n
- Authorization* header (HTTP), 63

B

- background* parameter for HTML tags, 83
- background processes, in JavaScript, 258
- backslashes (\) in URLs, browser acceptance of, 29
- backticks (`), as quote characters, 74, 111
- Bad Request status error (400), 55
- bandwidth, and XML, 123n
- Barth, Adam, 16, 177, 240, 241, 246, 257
- Base64 encoding, 50n
- basic credential-passing method, 63
- Bell-La Padula security model, 2, 4
- Berners-Lee, Tim, 9, 41, 69
 - and semantic web, 72–73
 - World Wide Web browser, 9
 - World Wide Web Consortium, 11
- `<bgsound>` tag (HTML), 84, 119
- binary HTTP, 257
- bitmap images, browser recognition of, 118
- blacklists
 - of HTTP headers in *XMLHttpRequest*, 147
 - malicious URLs, 236n
 - `_blank`, as link target, 80

- BMP file format, 83
- `<body>` tag (HTML), 83
- BOM (byte order marks), 208
- Breckman, John, 52n
- browser cache
 - information in, 59
 - poisoning, 60
- browser extensions and UI, 161
- browser-managed site permissions, 226–227
- browser market share, May 2011, 19
- browser-side scripts, 95–116
- browser wars, 10–11, 233
- buffer overflow, 265
- bugs, preventing classes of, 7
- Bush, Vannevar, 8
- byte order marks (BOM), 208

C

- cache. *See* browser cache
- Cache-Control* directive, 48, 59
- cache manifests, 257
- cache poisoning, 189, 263
- caching behavior, in HTTP, 58–60
- caching HTTP proxy, keepalive sessions and, 57
- Caja, 116
- Cake* (proposal), 257
- call stack, limiting size, 216
- callto*: scheme, 36
- `<canvas>` tag (HTML5), 183
- CAPTCHA, 184–185, 185n
- Cascading Style Sheets (CSS), 11, 12, 73, 83, 87–93
 - basic syntax, 88–90
 - character encoding, 91–92
 - interaction with HTML, 90
 - opacity* property, 179
 - parser resynchronization risks, 90–91
 - property definitions, 89
- case of tags, HTML vs. XML, 72
- `<![CDATA[...]]>` blocks, 72, 78, 250
- certificate authorities, 64
- certificates
 - extended validation, 65
 - warning dialog example, 66
- cf*: scheme, 36
- characters
 - delimiting, in URLs, 29
 - encoding in CSS, 91–92
 - encoding in filenames, 49–51
 - encoding in HTML, 76–78
 - encoding in JavaScript, 112–113
 - encoding in URLs, 31–35
 - printable, browser treatment of, 32

- reserved, 31–35
- unreserved, 32
- character sets
 - byte order marks and detection, 208
 - detection for non-HTTP files, 210–211
 - handling, 206–211
 - for headers, 49–51
 - inheritance and override, 209
 - markup-controlled, on subresources, 209–210
 - sniffing, 264
 - in URLs, 33
- @charset* (CSS), 89
- children objects in JavaScript, 108
- Chrome
 - autodetection of passive document types, 205
 - cached pages in, 37
 - characters in URL scheme name ignored by, 25
 - deleting JavaScript function, 103
 - and file extensions in URLs, 130
 - local file access, 160
 - modal dialogs for prompts, 219
 - navigation timing, 259
 - prerendering page, 258–259
 - printable characters in, 32
 - privileged JavaScript in, 161
 - and realm string, 63
 - and RFC 2047 encoding, 50
 - stored password retrieval, 228
 - SWF file handling without *Content-Type*, 199
 - time limits on continuously executing scripts, 215
 - WebKit parsing engine, 70n
 - window.open()* function and, 218
 - Windows Presentation Foundation plug-ins, 136
- chunked data transfers, 57–58
- clickjacking, 179, 180–181, 263
- click()* method, 218
- client certificates, 64–66
- client-server architecture, 17–18
- client-side data, 165
- client-side databases, 258
- client-side errors (400–499), 55–56
- client-side scripts, restricting privileges of
 - HTML generated by, 250–251
- cloud, 15
- Clover, Andrew, 184
- command injection, 265
- comments
 - in CSS syntax, 89
 - in XHTML and HTML, 72
- Common UNIX Printing System (CUPS), 152–153
- Common Vulnerability Scoring System (CVSS), 6–7
- Common Weakness Enumeration (CWE), 6
- complex selectors, in CSS, 88
- computer proficiency of user, 14
- conditionals, explicit and implicit, in HTML, 75–76
- conflicting headers, resolution of, 47–48
- CONNECT requests, 46, 54
- Connolly, Dan, 9
- content directives, on subresources, 204
- Content-Disposition* directive, 48, 84, 122
 - defensive uses, 203–204
 - NUL character and, 51
 - plug-in-executed code and, 204
 - user-controlled filenames in, 67
- content inclusion in HTML
 - hyperlinking and, 79–84
 - type-specific, 82–84
- Content-Length* header, 43, 52, 147
 - in keepalive sessions, 56–58
- content recognition, 197–211
- content rendering, plug-ins for, 127–138
- Content Security Policy (CSP), 242–245, 250, 253
 - criticisms of, 244–245
 - violations, 244
- content sniffing, 197–198, 205, 264
- Content-Type* directive, 49, 71, 84
 - application/binary*, 212
 - application/javascript*, 118
 - application/json*, 118, 202
 - application/mathml+xml*, 119
 - application/octet-stream*, 200–201, 212
 - charset* parameter, 206, 208
 - image/jpeg*, 118, 202, 205
 - image/svg+xml*, 124
 - logic to handle absence, 198–199
 - plug-ins and, 128, 204
 - slash-delimited alphanumeric tokens in, 199
 - special values, 200–201
 - text/css*, 118
 - text/html*, 124
 - text/plain*, 118, 156, 200–201, 204, 212
 - unrecognized, 202–203
 - and XML document parsing, 120
- control characters, JavaScript shorthand notation, 112
- cookie-authenticated text, reading, 181
- Cookie* header. *See* cookies
- cookie injection, 264

- cookies, 11, 257
 - deleting, 62
 - and DNS hijacking, 153
 - forcing, 264
 - limitations on third-party, 192–194
 - and same-origin policy, 150–151
 - security policy for, 149–153
 - semantics, 60–62
 - user data in, 67
- CORS. *See* Cross-Origin Resource Sharing (CORS)
- CR characters, stripping from HTTP headers, 45
- credential-passing methods, 63
- credentials, in URLs, 26
- CRLF (newline), 45
- cross-browser interactions, 16–17
- cross-document links, 8, 9
- cross-domain communications, and frame descendant policy, 176–178
- cross-domain content inclusion, 181–183
- cross-domain policy files, 155–156
- cross-domain requests, 236–239
- Cross-Origin Resource Sharing (CORS), 148, 236
 - current status, 239
 - non-simple requests and preflight, 238
 - request types, 236–237
 - security checks, 237–238
- cross-origin subresources, 183
- cross-site request forgery (XSRF, CSRF), 84, 190, 262
 - exploitation of flaws, 190
 - login forms and, 145–146
- cross-site script inclusion (XSSI), 104n, 262
- cross-site scripting (XSS), 71, 262
 - bugs, and password managers, 228
 - exploitation of flaws, 190
 - filtering, 251–252, 253
- crossdomain.xml* file, 155, 162
- CSP (Content Security Policy), 242–245, 250, 253
- CSRF (cross-site request forgery), 84, 190, 262
 - exploitation of flaws, 190
 - login forms and, 145–146
- CSS. *See* Cascading Style Sheets (CSS)
- CUPS (Common UNIX Printing System), 152–153
- currentStyle* API, 184
- CVSS (Common Vulnerability Scoring System), 6–7
- CWE (Common Weakness Enumeration), 6
- Cyrillic alphabet, homoglyphs in, 35

D

- daap*: scheme, 36
- data*: scheme, 37, 167–168
- data transfers, chunked, 57–58
- Date/If-Modified-Since* header pair, 59
- deceptive framing, 180
- dedicated workers, for background processes, 258
- default policy, CSP directive for, 243
- default ports, for protocols, overriding, 27
- DELETE method (HTTP), 53
- deleting
 - cookies, 62
 - JavaScript functions, 102–103
- delimiting characters, in URLs, 29
- denial-of-service (DoS) attacks, 214–219, 248, 264
- DeviceOrientation* API, 258
- dialog use restrictions, 218–219
- digest credential-passing method, 63
- Digital Rights Management (DRM), 131
- directory traversal, 265
- disable-xss-protection*, 242n
- `<div>` tag (HTML), 73
- DNS hijacking, and cookies, 153
- DNS labels, security mechanisms based on, 142n
- DNS names, in URLs, browser acceptance, 27
- DNS pinning, 142n, 190
- DNS rebinding, 142n, 189
- DNT* request header, 193
- `<!DOCTYPE>` directive, 71
- document.cookie* API (JavaScript), 61
- document.domain* property (JavaScript), 143–144
- document-level scrollbar, 180
- document* namespace, mapping HTML elements to, 110
- document* object (JavaScript), 108
- Document Object Model, 12, 108, 109–111, 142–146
- document rendering helpers, 130–131
- documents
 - changing location of existing, 174–178
 - script access to other, 111–112
- document type detection logic, 198–206
- Domain* parameter, for cookie, 61
- domains
 - hardcoded, 227
 - problems with restrictions, 151–152
- DOMService* mechanism, 158
- DoS (denial-of-service) attacks, 214–219, 248, 264

downloaded files, 205–206
drag-and-drop, 180
DRM (Digital Rights Management), 131
duplicate headers, resolution of, 47–48
Dutta, Sunava, 239

E

E4X. *See* ECMAScript for XML (E4X)
Earthlink, 153
ECMA (European Computer Manufacturers Association), 11, 96
ECMAScript, 96
 escape codes, 112
 strict mode, 104
ECMAScript for XML (E4X), 106–107
Eich, Brendan, 95
Electronic Frontier Foundation, 109
Eloquent JavaScript (Haverbeke), 97
<embed> tag (HTML), 83
 mixed content, 183
 src=..., 128
EMF file format, 83
encapsulating pseudo-protocols, 37–38
encoding schemes, for headers, 49–51
encryption, protocol-level, 64–66
enctype="text/plain", for <form> tag, 81
endless loop, 101, 215
ENQUIRE, 9, 10
<!ENTITY> directive, 76
entity encoding, in HTML, 76–78
error-handling rules, for certificates, 65–66
escaping reserved characters, in HTML, 71
escaping scheme, 91
Esser, Stefan, 209
ETag/If-None-Match header pair, 59
European Computer Manufacturers Association (ECMA), 11, 96
eval() function, 102
eval-script, 242n
Evans, Chris, 181, 182
EV SSL (Extended Validation SSL), 65
exception
 for *eval()* function, 102
 recovery in JavaScript, 100
execution time for scripts, 215–216
Expires directive, 48, 59
Expires parameter, for cookie, 61
explicit conditionals, in HTML, 75–76
expression(...) function (CSS), 89
Extended Validation SSL (EV SSL), 65
Extensible Application Markup Language (XAML), 134
extension matching, 202n

ExternalInterface.call() API, 133
External XML Entity (XXE) attack, 76

F

false positives, risk in XSS filtering, 251–252
fault tolerance, 11
feeds, 123–124
feed: scheme, 37
Felten, Ed, 193
file extensions, browser response to, 205
file formats. *See also* plug-ins
 audio and video, 119
 bitmap images, 118
 HTML. *See* HTML
 non-renderable, 124
 plaintext, 64, 85, 117–118
 XML. *See* XML
file inclusion, 265
file path, hierarchical, in URLs, 27–28
file: protocol, 159–160, 188
files, downloaded, 205–206
File Transfer Protocol (FTP), 26n, 205–206
filtering
 pop-up, 217–218
 reserved characters, in HTML, 71
Firefox (Mozilla), 13, 17
 and ActiveX, 137
 cached pages in, 37
 character set inheritance, 209
 CORS in, 239
 and credential portion of URLs, 26
 data: URLs in, 168
 DNT request header, 193
 entity names, 77
 external content directives, 90
 Gecko parsing engine, 70n
 history.pushState() API, 256
 javascript: URLs in, 169
 local file access, 160
 modal dialogs for prompts, 219
 multiple cookies for, 62
 printable characters in, 32
 privileged JavaScript in, 161
 prompt displayed when saving *Content-Type: image/jpeg* document, 205
 redirects to *about:blank*, 166
 and RFC 2047 encoding, 50
 RSS and Atom renderers for, 124
 same-origin policy loopholes, 185
 stored password retrieval, 228
 Strict Transport Security support, 248
 SWF file handling without
 Content-Type, 199

- Firefox (*continued*)
 - time limits on continuously executing scripts, 215
 - UTF-8 text in, 50
 - Windows Presentation Foundation
 - plug-ins, 136
 - Worker API, 258
- firefoxurl*: protocol, 17, 36
- Flash applets, 11
- fonts
 - CSP directive for, 243
 - Flash programs enumeration of, 132
- Forbidden status code (403), 56
- forecasting, statistical, 6
- format-string vulnerability, 266
- form-based password managers, 227–229
- form feed character, in HTML tag, 74
- forms, 80–82
- Found status code (302), 55
- fragment ID, in URLs, 28–29
- frame-ancestors* directive, 243
- framebusting, 264
- frame descendant policy, and cross-domain communications, 176–178
- frames, 82
 - disabling navigation descendant model, 230–231
 - hijacking risks, 175–176
 - name* attribute of, 175
 - sandboxed, 245–247
 - unsolicited, 178–181
 - window interactions, 174–181
- frame-src* directive, 243
- From-Origin* header, 240
- FTP (File Transfer Protocol), 26n, 205–206
- ftp*: scheme, 36
- full-screen mode, proposals for, 259
- fully qualified absolute URLs, 24
- fully restricted URL scheme, 188
- functional notation, in CSS, 89
- functions
 - JavaScript, overriding, 102–103
 - resolution for JavaScript, 98–99

G

- Gabrilovich, Evgeniy, 35
- Gecko parsing engine, 70n
- Generalized Markup Language (GML), 8–9
- geolocation data, 226
- geolocation discovery, 258
- geolocation-sharing prompts, 223
- getComputedStyle* API, 184
- getElementById()* function, 109

- getElementsByName()* function, 109
- GET method (HTTP), 42, 52, 58, 80–81
- GetRight download utility, 137
- getters, in JavaScript, 103
- getURL()* function, 133
- GIFAR vulnerability, 129
- GIF file format, 83, 129
- GML (Generalized Markup Language), 8–9
- Gontmakher, Alex, 35
- Gonzalez, Albert, 5n
- gopher*: scheme, 36
- Gosling, James, 134
- GPS data, 226n
- Grossman, Jeremiah, 179
- Guninski, Georgi, 176

H

- Hansen, Robert, 179
- hardcoded domains, 227
- Haverbeke, Marijn, *Eloquent JavaScript*, 97
- HDP file format, 83
- header injection, 45, 239, 262
- headers
 - character set and encoding schemes, 49–51
 - Content Security Policy encoded in, 242
 - in HTTP requests, 43
 - resolution of duplicate or conflicting, 47–48
 - semicolon-delimited values, 48–49
- HEAD request (HTTP), 53
- hexadecimal notation, 77, 112
- hierarchical file path, in URLs, 27–28
- history* object (JavaScript), 108
- history.pushState()* API, 256
- Hodges, Jeff, 248
- homoglyphs, in Cyrillic alphabet, 35
- Host* request header, 43
- hostnames
 - extra periods, and cookie-setting algorithms, 159
 - non-fully qualified, 159
- HTML (Hypertext Markup Language), 9, 69–86
 - basic concepts, 70–73
 - case of tags, 72
 - converting to plaintext, 85
 - CSS interaction with, 90
 - document misidentified as, 198
 - document parsing modes, 71–72
 - embedded in feed formats, 124
 - entity encoding, 76–78
 - explicit and implicit conditionals, 75–76

- HTTP integration semantics, 78–79
 - hyperlinking and content inclusion, 79–84
 - in-browser sanitizers, 250–251
 - mapping elements to *document* namespace, 110
 - parser behavior, 73–76
 - tag interactions, 74–75
 - type-specific content inclusion, 82–84
 - version 4, 12
 - version 5, 70, 119, 131
 - HTTP (HyperText Transfer Protocol), 9, 41–67
 - authentication, 62–63
 - basic syntax, 42–51
 - binary, 257
 - caching behavior, 58–60
 - cookie semantics, 60–62
 - downgrade, 264
 - history, 41–42
 - HTML integration semantics, 78–79
 - newline handling, 45
 - proxy requests, 46–47
 - request types, 52–54
 - semantics battle, 72–73
 - simultaneous connections, 216
 - version 0.9, 42–43, 44
 - version 1.0, 42, 43, 44, 48, 59
 - version 1.1, 42–43, 45, 48, 57, 198
 - httponly* flag, for cookie, 61, 150
 - http:* scheme, 36
 - HTTPS, 65
 - documents, 138n, 183
 - downgrade risks, 248
 - https:* scheme, 36
 - hyperlinking, and content inclusion, 79–84
 - Hypertext Markup Language (HTML).
 - See HTML (Hypertext Markup Language)
 - HyperText Transfer Protocol (HTTP).
 - See HTTP (HyperText Transfer Protocol)
- I**
- IANA (Internet Assigned Numbers Authority), 24, 152
 - ICO file format, 83
 - IDNA (Internationalized Domain Names in Applications), 34–35
 - IETF (Internet Engineering Task Force), 11
 - If-Modified-Since* header, 59
 - If-None-Match* header, 59
 - `<iframe>` tag (HTML), 82, 176, 209, 245–247
 - image/jpeg* document type, 118, 202, 205
 - images
 - bitmap, 118
 - in HTML, 83
 - risk of content sniffing on, 202
 - Scalable Vector Graphics (SVG), 83, 121–122
 - image/svg+xml* document type, 124
 - `` tag (HTML), 83
 - src* parameter, 181
 - for SVG images, 122
 - implicit caching, 59
 - implicit conditionals, in HTML, 75–76
 - @import*, in CSS, 89–90
 - IndexedDB* design, 258
 - indicator of hierarchical URLs, 25–26
 - information security, 1–8
 - inheritance, for *vbscript:* scheme, 169–170
 - inline-script* setting, 242n
 - innerHTML* property, 110–111
 - innerHTMLStaticHTML* API, 251
 - integer overflow, 266
 - Interactive Voice Response (IVR) systems, 236
 - interconnected systems, losses in, 5
 - internal networks, access to, 189–190
 - Internal Revenue Service, 231
 - Internal Server Error (500), 56
 - International Organization for Standardization (ISO), 11
 - Internationalized Domain Names in Applications (IDNA), 34–35
 - Internet Assigned Numbers Authority (IANA), 24, 152
 - Internet Engineering Task Force (IETF), 11
 - Internet Explorer, 10, 11–12
 - ActiveX and, 137
 - and `<% ... %>` blocks, 75
 - `\` (backslash) in URLs, 29
 - acceptance of backtick as quote, 74
 - characters in URL scheme name
 - ignored by, 25
 - clickjacking, 182
 - content sniffing, 202
 - cookies, 149
 - data:* URLs in, 168
 - delete attempt of JavaScript function, 103
 - extension matching, 202
 - fallback display, 118
 - and file extensions in URLs, 130
 - frames, 177
 - JavaScript in, 96
 - JSON.parse()* function alternative, 104
 - local file access, 160
 - markup controlled charset on, 209

- Internet Explorer (*continued*)
 - and multiline headers, 45
 - multiline string literals support, 91
 - non-recognition of vertical tab, 112
 - NUL character and, 73, 74
 - origin check and port number, 142
 - printable characters in, 32
 - proprietary *security-restricted* parameter, 246
 - redirects to *about:blank*, 166–167
 - and RFC 2047 encoding, 50
 - same-origin policy and, 143n, 185
 - Silverlight and, 134
 - stored password retrieval, 228
 - SWF file handling without *Content-Type*, 199
 - text/plain* document type, 200–201
 - third-party cookies blocking, 193
 - time limits on continuously executing scripts, 215
 - Trident parsing engine, 70n
 - VBScript, 96, 114
 - window.open()* function and, 218
 - Windows Presentation Foundation plug-ins, 136
 - XDomainRequest* approach to, 148
 - XSS-detection logic, 251
 - Zone.Identifier* metadata, 231
 - zone model, 229–231
 - Internet Information Server, and *Host* headers, 47
 - Internet service providers, 153
 - Internet zone, for Internet Explorer, 230
 - interstitials, 218
 - intrusions
 - escalation of, 5
 - nonmonetary costs, 5
 - Invisible Gorilla experiment, 223
 - IP addresses, and cookies, 158
 - ISO (International Organization for Standardization), 11
 - ISO-8859-1 (Western European code page), 50
 - itms:* scheme, 36
 - itpc:* scheme, 36
 - IVR (Interactive Voice Response) systems, 236
- J**
- Jackson, Collin, 16, 177, 184, 240
 - jar:* scheme, 37
 - Java, 134–135, 157–158
 - Java Runtime Environment (JRE), 135
 - JavaScript, 10, 11n, 83, 95–107
 - character encoding in, 112–113
 - code and object inspection capabilities, 101–102
 - code execution, 100
 - code inclusion modes and nesting risks, 113–114
 - document.domain* property, 143–144
 - Document Object Model, 12, 108, 109–111
 - embedded in PDF documents, 130
 - execution order control, 100–101
 - labeled statements support, 105n
 - MIME type, 118n
 - Netscape and, 95–96
 - runtime environment for, 102–104
 - script processing model, 97–100
 - setters and getters, 103
 - standard object hierarchy, 107–112
 - variable declaration, 99
 - and WML Script (WMLS), 123
 - JavaScript Object Notation (JSON), 104–106, 112
 - javascript:* scheme, 37, 169–170
 - Jobs, Steve, 131
 - JPEG file format, 83
 - JScript, 11n
 - JScript.Encode, 113n
 - JObject* mechanism, 158
 - JSON (JavaScript Object Notation), 104–106, 112
 - JSONP (JSON with padding), 106n, 245
 - JSON.parse()* function, alternatives, 104
- K**
- Kaminsky, Dan, 153
 - katakana, 33
 - keepalive sessions, 56–57, 216
 - keystroke redirection, 180
 - Kinugawa, Masato, 210
- L**
- language* parameter, for `<script>` tag, 113n
 - Lessig, Lawrence, 192
 - LF (newline), HTTP quirks in handling, 45
 - LFI (local file inclusion), 265
 - Lie, Wium, Håkon, 87
 - `<link rel=stylesheet>` directive, 88
 - `<link rel=stylesheet href=...>` tag, 181
 - LiveScript, 95
 - livescript:* scheme, 37

loadPolicyFile() method, 155–156
 local file inclusion (LFI), 265
 local files, access issues, 159–160
 local intranet zone, for Internet Explorer, 229
 local machine zone, for Internet Explorer, 229
localhost, danger of, 152–153
localStorage object (JavaScript), 148
location.hash, 256
 location headers, sending user-controlled, 67
location.host property, 173
location object (JavaScript), 108, 153–154
 location of documents, changing, 174–178
 login forms, autocompletion by browsers, 228
 lookup functions, in Document Object Model, 109
 loopback interfaces, 152n
 Lynx, 10

M

Macromedia Flash, 132
mailto: protocol, 25, 36, 256
 mail user agent (MUA), 203n
 malicious sites, blacklist-driven attempts to block, 226
 managed code, 134n
 Mark of the Web (MotW), 204, 231
 markup filter for user content, 86
 mashups, 176
 MathML (Mathematical Markup Language), 72, 122
Math.random() function, 109
max-age parameter
 for cookie, 61
 for STS record, 248
 media capture, 259
 Memex, 8
 memory pointers, 266
 memory use restrictions for scripts, 215–216
 <meta> directive, 206, 208
 <meta http-equiv=> directive, 78–79
 meta-policies, for Flash, 156–157
mhtml protocol, 38
 Microsoft. *See also* Internet Explorer
 descendant policy development, 177
 .NET Framework with XPAB
 plug-ins, 136
 objections to CORS, 239
 Sun suit over Java virtual machine, 135n

Threats Against and Protection of Microsoft's Internal Network, 5n
 Windows operating system, 10
 Microsoft Office, 130
 Microsoft Silverlight, 119, 134, 157
 MIME (Multipurpose Internet Mail Extensions), 43n, 81n
 malformed types, 199
 mapping types to plaintext, 118
 for plug-ins, 128
 specialized for content in sandboxed frame, 247
 Mitchell, John C., 177, 240
 mixed content, 183, 262–263
mmst: scheme, 36
mmsu: scheme, 36
 Mocha language, 95
mocha: scheme, 37
 modal behavior of dialogs, 218–219
 Montulli, Lou, 60
 Mosaic, 10. *See also* Netscape
 MotW (Mark of the Web), 204, 231
 mouse cursors, redefining, 89n
 Moved Permanently status code (301), 55
 Mozilla Firefox. *See* Firefox (Mozilla)
 Mozilla specification, 242
msbd: scheme, 36
MsgBox (VBScript), 114
 MUA (mail user agent), 203n
 multiline headers, support for, 45
 multiline string literals
 Internet Explorer support, 91
 in JavaScript, 113
 multimedia playback, 130
 Multipurpose Internet Mail Extensions (MIME). *See* MIME (Multipurpose Internet Mail Extensions)
 My computer zone, for Internet Explorer, 229

N

name attribute, of frames, 175
 named entities, 76
 namespace in JavaScript, 107
name: value pairs, in HTTP requests, 43
name=value pairs
 cookies for storing, 60
 for forms, 81
 National Science Foundation, backbone network, 10
 Naval Research Laboratory, 3–4
navigateToURL() function, 133

- navigation
 - to sensitive schemes, 188
 - timing, 259
- navigator.device.capture* API, 259
- navigator.geolocation.getCurrentPosition()* API, 258
- navigator* object (JavaScript), 108
- navigator.registerProtocolHandler()* API, 256
- Negotiate authentication method, 63
- .NET runtime, 135
- Netflix, 134
- Netscape
 - cookie specification, 151–152
 - and JavaScript, 95–96
 - and same-origin policy, 142
- Netscape Navigator, 11
- network fenceposts, 264
- networking, HTTP-less, 257
- New York Times*, 192
- newline, HTTP quirks in handling, 45
- news:* scheme, 36
- NLS, 9
- nntp:* scheme, 36
- no-cache* value, for *Cache-Control* header, 59
- No Content status code (204), 54
- noncanonical encodings, 32n
- nonencapsulating pseudo-protocols, 37
- non-HTTP resources
 - character set detection for, 210–211
 - proxies allowing requests for, 46
- non-renderable file types, 124
- non-US-ASCII text, in URLs, 32–35
- no-store* value, for *Cache-Control* header, 59
- Not Found status code (404), 56
- Not Modified status code (304), 55, 59
- Notification API, 259
- NTLM authentication method, 63
- NUL character, and HTTP headers, 51
- NUL-containing strings,
 - JavaScript and, 109

O

- Object Linking and Embedding (OLE), 136
- <object>* tag, 83, 84
 - data=...*, 128
 - mixed content, 183
- octal character codes, JavaScript support, 112
- Ogg Theora, 119
- OK status code (200), 54
- OLE (Object Linking and Embedding), 136
- onbeforeunload* dialog, 219n

- onerror* handler, on ** tag, 184
- onerror* parameter, 74
- onkeydown* event (JavaScript), 180
- onload* handler, to measure load time for document, 184
- onmousemove* events, 222
- opacity* property (CSS2), and JavaScript code, 179
- opener.window.focus()* function, 217n
- OpenGL-based 3D graphics, 131n
- open redirection, 263
- Opera, 10
 - data:* URLs in, 168
 - deleting JavaScript function, 103
 - and file extensions in URLs, 130
 - history.pushState()* API, 256
 - local file access, 160
 - modal dialogs for prompts, 219
 - and multiline headers, 45
 - period-counting problem in, 159
 - Presto parsing engine, 70n
 - printable characters in, 32
 - redirects to *about:blank*, 167
 - Refresh* redirection to *javascript:*, 170
 - and RFC 2047 encoding, 50
 - RSS and Atom renderers for, 124
 - stored password retrieval, 228
 - SWF file handling without *Content-Type*, 199
 - Worker* API, 258
- OPTIONS method (HTTP), 53
- Origin* header, 240–241
- origin inheritance, 165–171
 - about:blank* document, 166–167
 - for *javascript:* scheme, 169–170
- origins
 - ambiguous or unexpected, 158–161
 - attempts to broaden, 143
- Ormandy, Tavis, 152
- outerHTML* property, 110–111
- overwriting cookie, 62

P

- P2P networking, 257
- P3P (Platform for Privacy Preference), 193
- Panopticlck, 109
- parallel HTTP connection design, 216
- <param>* tag (HTML), for plug-ins, 128
- _parent*, as link target, 80
- parsing
 - behavior fundamentals, 73–76
 - JavaScript, 97–98
 - modes for HTML documents, 71–72
 - resynchronization risks, 90–91

- parsing engines, for browsers, 70n
- Partial Content status code (206), 54
- partly restricted URL scheme, 188
- passive multimedia, CSP directive for, 243
- password
 - in credentials portion of URLs, 26
 - form-based managers, 227–229
 - methods for passing, 63
- Path* parameter, for cookie, 61
- path* value, for cookie, 149–150
- payload inspection, by Internet Explorer, 202
- PDF documents 130–131
- percent encoding, 31
- percent sign (%), for character encoding, 31
- per-host connection limit, 216
- period (.), hostnames with, and cookie-setting algorithms, 159
- permissions, browser- and plug-in-managed, 226–227
- permitted-cross-domain-policies* parameter, for *crossdomain.xml* file, 162
- persistent workers, for background processes, 258
- Petkov, Petko D., 131
- phishing, 176n
- plaintext
 - converting HTML to, 85
 - as file format, 117–118
 - for HTTP session information, 64
- `<plaintext>` tag (HTML), 72
- Platform for Privacy Preference (P3P), 193
- plug-ins, 10–11
 - ActiveX, 129, 136–137
 - Adobe Flash. *See* Adobe Flash
 - application frameworks as basis, 131–136
 - content, 83
 - for content rendering, 127–138
 - CSP directive for, 243
 - document rendering helpers, 130–131
 - invoking, 128–130
 - Microsoft Silverlight, 119, 134, 157
 - for PDF documents, 130–131
 - perils of content-type handling, 129–130
 - protocols claimed by, 36–37
 - security rules, 153–158
 - site permissions management, 226–227
 - Sun Java, 134–135, 157–158
 - XML browser applications (XBAP), 135–136
- PNG file format, 83
- pointers, management vulnerabilities, 266
- poisoned browser cache, on trusted network, 60
- pop-under, 217
- pop-up filtering, 217–218
- ports
 - default, for protocols, overriding, 87
 - prohibited, 190–192
- positioning windows, 219–222
- postMessage(...)* API, 144–145, 258
- POST method (HTTP), 52, 81
- postponing JavaScript execution, 101
- Pragma: no-cache* request header, 59
- prerendering web page, 258–259
- presentation, HTML tags for, 73
- PresentationHost.exe*, 135
- pressed key, examining code of, 180
- Presto parsing engine, 70n
- printable characters, browser treatment of, 32
- privacy-related side channels, 184–185
- private browsing modes, 249, 253
- private* value, for *Cache-Control* header, 59
- privileges, site, 225–234
- prohibited ports, 190–192
- properties, definitions in CSS, 89
- proposals
 - content-level, 258–259
 - I/O interfaces, 259
 - URL- and protocol-level, 256–257
- protocol-host-port tuple, 142, 241
- protocol-level information
 - encryption, 64–66
 - preserving, 78
- protocol-level proposals, 256–257
- protocols
 - claimed by third-party applications, 36–37
 - default ports for, overriding, 27
 - registration, 256
 - in URL scheme name, 24
- proxy-originating error responses, browser processing, 47
- proxy requests, 46–47
- pseudo-functions (CSS), 89
- pseudo-protocols
 - encapsulating, 37–38
 - nonencapsulating, 37
- pseudo-URLs, 23, 24, 165
 - restricted, 170–171
 - and same-origin policy, 161
- public key cryptography, 64, 64n
- Public Suffix List, 159
- public* value, for *Cache-Control* header, 59
- public Wi-Fi networks, and HTTP caching risk, 60
- Punycode, 34
- purging browser cache, 60
- PUT request (HTTP), 53

Q

- query string in URLs, 28
- QuickTime (Apple), 119, 130, 132
- quoted characters, in HTML, 71, 74
- quoted-printable encoding scheme, 50n
- quoted-string* syntax, 48–49
 - and cookies, 62
 - for CSS property values, 89

R

- race conditions, in JavaScript, 101
- raw text, for CSS property values, 89
- Really Simple Syndication (RSS), 123
- realm string, 62
- RealNetworks RealPlayer, 130, 132
- redirect headers, sending user-controlled, 67
- Redirection status codes (300–399), 55
- Referer* header, 43, 51
 - alternative to, 240
 - leakage, 263
- relative URLs, 24
 - vs. absolute, 25
 - input filters, 40
 - resolution of, 38–39
- remote file inclusion (RFI), 265
- Request for Comments (RFC). *See* RFC (Request for Comments)
- request headers, in HTTP, 43
- request types
 - form-triggered, 80–82
 - HTTP, 52–54
- reserved characters, in HTML, 31–35, 71
- resource exhaustion attacks, 214
- response codes, server, 54–56
- response splitting, 45
- Restricted sites zone, for Internet Explorer, 229–230
- revalidation, 59
- RFC (Request for Comments)
 - 1630
 - on query string format, 28
 - on reference parser, 25–26
 - 1738, on URLs, 24, 25
 - 1866, on HTML 2.0, 69
 - 1945
 - on HTTP, 42
 - and TEXT token, 50
 - 2046, on *application/octet-stream*, 200
 - 2047, for non-ISO-8859-1 string format, 50
 - 2109, on cookies, 60, 61, 62
 - 2183, on *Content-Disposition* header, 203
 - 2368, on query string format, 28

- 2616, 44
 - on GET requests, 58
 - on HTTP, 42
 - on resolving ambiguities, 47
 - status codes for server response, 54
 - on URLs, 24
- 2617, on authentication, 62
- 2818, on encapsulation, 64
- 2965, on *Cookie2*, 60
- 3490, 34
- 3492, 34
- 3986, 24, 25, 33
- 4627, on JSON, 104
- 4918, on WebDAV, 54
- 6265, on cookies, 61
- browser permissions to examine payload, 198
 - on HTTP, 48
- RFI (remote file inclusion), 265
- rgb(...)* pseudo functions (CSS), 89
- Riley, Chris John, 203
- Rios, Billy, 129
- risk management, 4–6
- root object in JavaScript, 107
- Ross, David, 251
- rotate(...)* pseudo functions (CSS), 89
- RSS (Really Simple Syndication), 123
- rtsp:* scheme, 36
- runtime environment, for JavaScript, 102–104

S

- Safari (Apple), 13
 - and credential portion of URLs, 26
 - deleting JavaScript function, 103
 - hiding address bar, 221
 - and multiline headers, 45
 - and realm string, 63
 - RSS and Atom renderers for, 124
 - SOP bypass flaws, 142n
 - stored password retrieval, 228
 - SWF file handling without
 - Content-Type*, 199
 - text/plain* document type, 200–201
 - third-party cookies, 193
 - time limits on continuously executing scripts, 215
 - WebKit parsing engine, 70n
- safeInnerHTML* API, 251
- same-origin policy mechanism, 16
 - cookies impact on, 150–151
 - for Document Object Model, 142–146
 - limitations, 173–186
 - loopholes, 185

- and pseudo-URLs, 161
- for web storage, 148
- for *XMLHttpRequest* API, 146–148
- sandbox* directive, 244
- sandboxed frames, 245–247, 250, 253
 - scripting, forms and navigation restrictions, 247
 - synthetic origins, 247
- sanitization
 - in-browser HTML, 250–251
 - of tags, 76
- Scalable Vector Graphics (SVG), 83, 121–122
- scale(...)* pseudo functions (CSS), 89
- schemes
 - current list of valid names, 24
 - input filters, 40
 - name in URLs, 24–25
 - navigation to sensitive, 188
- Schwab, Charles, 230–231
- screen* object (JavaScript), 108
- script-nonce* directive, 244
- scripts, 83
 - access to other documents, 111–112
 - browser-side, 95–116
 - connection limits, 216–217
 - dialog use restrictions, 218–219
 - execution time and memory use restrictions, 215–216
 - pop-up filtering, 217–218
 - rogue, 213–224
 - specifying charset, 209
- script-src* directive (CSP), 242
- `<script>` tag (HTML), 72
 - JSON and, 104–105
 - language* parameter, 113n
 - parsing and, 98
 - src* parameter, 181
- `<script>` tag (XHTML), 78
- scrollbar, document-level, 180
- Secure attribute, for cookie, 61
- secure* cookies, 150, 162
- security
 - actions subject to checks, 141
 - definition, 2–4
 - new and upcoming features, 235–253
 - practical approaches, 7–8
 - quality assurance, 7
- Security.allowDomain(...)* method, for Flash, 155
- security dialogs, attacks on, 222–223
- security engineering cheat sheet
 - building web applications on internal networks, 195
 - Content Security Policy (CSP), 253
 - converting HTML to plaintext, 85
 - cross-domain communications in JavaScript, 162, 186
 - cross-domain resources, 186
 - cross-domain *XMLHttpRequest* (CORS), 253
 - data:* and *javascript:* URLs, 172
 - decoding parameters received through URLs, 40
 - embedding plug-in-handled active content from third parties, 162
 - enabling plug-in-handled files, 138
 - filtering user-supplies CSS, 93
 - generating documents with partly attacker-controlled contents, 212
 - generating HTML documents with attacker-controlled bits, 85
 - good practices for all websites, 212
 - hosting user-generated files, 212
 - hosting XML-based document formats, 125
 - hosting your own plug-in-executed content, 163
 - hygiene for all HTML documents, 85
 - interacting with browser objects on client side, 115
 - launching non-HTTP services, 195
 - loading remote scripts, 115
 - loading remote stylesheets, 93
 - markup filter for user content, 86
 - non-HTML document types, 125
 - parsing JSON from server, 115
 - permitting user-created `<iframe>` gadgets on site, 224
 - private browsing modes, 253
 - putting attacker-controlled values into CSS, 93
 - relying on HTTP cookies for authentication, 162
 - requesting elevated permissions within web application, 232
 - sandboxed frames, 253
 - security hygiene for all websites, 186
 - security policy hygiene for all websites, 162
 - security-sensitive UIs, 224
 - sending user-controlled location headers, 67
 - sending user-controlled redirect headers, 67
 - serving plug-in-handled files, 138
 - Strict Transport Security, 253
 - third-party cookies for gadgets or sandboxed content, 195
 - toStaticHTML()* API, 253

- security engineering cheat sheet
 - (*continued*)
 - URL input filters, 40
 - URLs constructed based on user input, 40
 - user-controlled filenames in *Content-Disposition* headers, 67
 - user-controlled scripts, 116
 - user data in HTTP cookies, 67
 - user-specified class values on HTML markup, 93
 - user-supplied data inside JavaScript blocks, 115
 - writing browser extensions, 163
 - writing plug-ins or extensions recognizing privileged origins, 232
 - XDomainRequest*, 253
 - XSS filtering, 253
- security model extension frameworks, 236–241
 - cross-domain requests, 236–239
 - XDomainRequest*, 239–240
- security model restriction frameworks, 241–249
- See Other status code (303), 55
- selector suffixes, in CSS, 88
- _self*, as link target, 80
- self-closing tag syntax, 72
- semantic web, 72–73
- semicolon (;), as delimiter
 - in HTTP headers, 48–49
 - in URLs, 29
- server address, in URLs, 26–27
- server port, in URLs, 27
- server response codes, 54–56
- server-side code, common problems
 - unique to, 265–266
- server-side errors (500–599), 56
- Service Unavailable error (503), 56
- sessionStorage* object (JavaScript), 148
- Set-Cookie* headers, 61
- setters, in JavaScript, 103
- SGML (Standard Generalized Markup Language), 9
- shared workers, for background processes, 258
- Shockwave Flash, 132
- SHODAN, 203
- showModalDialog()* method, 217
- shhttp*: scheme, 36
- Simple Mail Transfer Protocol (SMTP), 27, 44, 190
- sip*: scheme, 36
- <site-control permitted-cross-domain-policies="..">* parameter, 157
- site privileges, 225–234
 - browser- and plug-in-managed permissions, 226–227
- skew(...)* pseudo functions (CSS), 89
- SMTP (Simple Mail Transfer Protocol), 27, 44, 190
- social engineering attacks, 32n
- software, difficulty analyzing behavior of, 3
- * tag (HTML), 73
- SPDY (Speedy), 257
- Spyglass Mosaic, 10
- SSL, warnings appearance, 66
- Standard Generalized Markup Language (SGML), 9
- statistical forecasting, 6
- Sterne, Brandon, 242
- Stone, Paul, 180
- Strict Transport Security (STS), 248–249, 253
- strict XML mode, 72
- stylesheets
 - CSP directive for, 243
 - specifying charset, 209
- <style>* tag (HTML), 72
- <style>* tag (XHTML), 78
- subframes, CSP directive for, 243
- subresources
 - cross-origin, 183
 - markup-controlled charset on, 209–210
- Sun Java, 134–135, 157–158
- Sun Microsystems, 129
- SVG (Scalable Vector Graphics), 83, 121–122
- <svg>* tag (HTML5), 122
- synchronous *XMLHttpRequest*, 146
- syntax-delimiting characters, in URLs, 31
- “syntax error” message, retrieved file snippet in, 181

T

- <table>* tag (HTML), 83
- tags, in HTML, 70
 - handling those not closed before end of file, 75
 - interactions, 74–75
 - sanitization, 76
- target* parameter, for ** tag (HTML), 79, 174–175
- taxonomy, 6–7
- TCP/IP, HTTP and, 42
- TCP (Transmission Control Protocol), 42n
- connections via XMLHttpRequest, 156
- list of prohibited ports, 190–192
- Temporary Redirect status code (307), 55

- testing, for Internet Explorer use, 112
- text/css* document type, 118
- text/csv* document type, 198
- text/html* document type, 124
- text message, sending to window with
 - valid JavaScript handle, 144
- text/plain* document type, 118, 156, 200–201, 204, 212
- TEXT token, 50
- <*textarea*> tag (HTML), 72, 111
- third-party applications, protocols claimed by, 36–37
- third-party cookies, limitations, 192–194
- threat evolution, 14–18
 - cloud, 15
 - nonconvergence of visions, 15–16
 - user as security flaw, 14–15
- Threats Against and Protection of Microsoft's Internal Network* (Microsoft), 5n
- three-step TCP handshake, 56
- TIFF file format, 83
- timer, in JavaScript, 101
- timing attacks, on user interfaces, 222–223
- TLS (Transport Layer Security), 64
- _top*, as link target, 80
- top-level domains, 152
- toSource()* method (JavaScript), 101
- toStaticHTML()* API, 250–251, 253
- toString()* method (JavaScript), 101
- TRACE method (HTTP), 53
- tracking, unscrupulous online, 193
- tragedy of the commons dilemma, 3
- Transfer-Encoding: chunked scheme, 58
- Transmission Control Protocol (TCP). *See* TCP (Transmission Control Protocol)
- Transport Layer Security (TLS), 64
- Trident parsing engine, 70n
- Trusted sites zone, for Internet Explorer, 229
- Turing, Alan, 3n
- type* parameter, for plug-in tag, 128

U

- UI spoofing attacks, and Flash, 132
- unauthenticated requests, by browser, 62
- Unauthorized status error (401), 55, 62
- unhandled exception, in JavaScript, 100
- Unicode, 33
 - decimal $\&\#number$; notation for, 77
 - escaping method based on, 113
 - JavaScript support, 112
 - whitespace, 74n

- Uniform Messaging Policy, 240
- Uniform Resource Locators (URLs), 23–40
 - browser processing, 29–31
 - common schemes, 36–38
 - constructing based on user input, 40
 - encoding, 31
 - encoding data in fragment
 - identifiers, 144n
 - fully qualified absolute, 24
 - hiding with encapsulating protocols, 38
 - navigation based on tiers of schemes, 188
 - resolution of relative, 38–39
 - structure, 24–31
 - credentials, 26
 - fragment ID, 28–29
 - hierarchical file path, 27–28
 - indicator of hierarchical URLs, 25–26
 - query string, 28
 - scheme name, 24–25
 - server address, 26–27
 - server port, 27
- UniformRequest* API, 240
- University of Illinois, 10
- Unix services, listener process, 216n
- unreserved characters, in HTML, 32
- unrestricted URL scheme, 188
- URLs (Uniform Resource Locators). *See* Uniform Resource Locators (URLs)
- URL-handling APIs, 133
- URL-level proposals, 256–257
- url(...)* pseudo-functions (CSS), 89
- user
 - browsing habits, *Referer* header and, 51
 - collecting information about
 - interaction, 184
 - as security flaw, 14–15
 - URL construction based on input, 40
- User-Agent* request header, 43
- user content, markup filter for, 86
- user-controlled filenames in *Content-Disposition* headers, 67
- user data in HTTP cookies, 67
- user interfaces
 - browser extensions and, 161
 - notifications, 259
 - timing attacks on, 222–223
- username, in credentials portion of URLs, 26
- UTF-7 charset, 78
- UTF-8 charset, 33, 206
 - in HTTP headers, 50
- UTF-16 charset, 78, 206
- UTF-32 charset, 78

V

- valid scheme names, current list, 24
- variables, declaration in JavaScript, 99
- VBScript, 96
- vbscript*: scheme, 37, 169–170
- vertical tab, in HTML tag, 74
- `<video>` tag (HTML5), 84, 119, 131
- view-cache*: scheme, 37
- View > Encoding* menu, 209
- view-source*: scheme, 37
- Visual Basic, 10, 114, 130
- VoiceXML, 236

W

- W3C (World Wide Web Consortium), 12, 70
- w3m, 10
- WAP (Wireless Application Protocol suite), 123
- WBXML, 123n
- WDP file format, 83
- Web, the. *See* World Wide Web
- web 2.0, 12–13
- web applications
 - design issues, 263–265
 - vulnerabilities specific to, 262–263
- WebDAV, 54
- WebGL, 131, 131n
- Web Hypertext Application Technology Working Group (WHATWG), 13
- WebKit parsing engine, 70n, 242
 - character set inheritance, 209
 - CORS in, 237, 239
 - data*: URLs in, 168
 - history.pushState()* API, 256
 - Refresh* redirection to *javascript*:, 170
 - Strict Transport Security support, 248
 - Worker* API, 258
 - XSS-detection logic, 251
- web page, prerendering, 258–259
- web storage, same-origin policy mechanism for, 148
- WebRTC, 257
- WebSocket API, 257
- WebSQL* API, 258
- Western European code page (ISO-8859-1), 50
- WHATWG (Web Hypertext Application Technology Working Group), 13
- whitelists, 226
- whitespace, 74, 92
- window.alert()* API, 218

- window.blur()* function, 217n, 220
- window.confirm()* API, 218
- window.createPopup()* API, 222
- window.focus()* method, 220
- window handles, 175
- window.moveTo()* method, 220
- window.name* property, of frames, 175
- window.notifications* API, 259
- window.open()* function, 111, 174–175, 217, 217n, 219, 222
- window.print()* API, 218
- window.prompt()* API, 218
- window.resizeTo()* method, 220
- windows
 - anonymous, 175
 - creating new in browser, 217
 - and frame interactions, 174–181
 - positioning, 219–222
- window.showModalDialog()* API, 217
- Windows Media Player, 119, 130, 132
- Windows operating system, 10, 13
- window splicing, 220–221
- Windows Presentation Foundation, 134, 136
- Wireless Application Protocol suite (WAP), 123
- Wireless Markup Language (WML), 123
- WMF file format, 83
- WML Script (WMLS), and JavaScript, 123
- WML (Wireless Markup Language), 123
- Worker* API, 258
- World Wide Web
 - browser wars, 10–11, 233
 - history, 8–13
 - threat of hostile takeover, 131
- World Wide Web Consortium (W3C), 12, 70
 - creation of, 11
 - Microsoft and, 239
- worms, 12
- WWW-Authenticate* header, 62, 63
- wyciwyg*: scheme, 37

X

- XAML (Extensible Application Markup Language), 134
- Xanadu, 9
- XBAP (XML browser applications), 135–136
- XBL bindings, 89–90
- X-Content-Type-Options* header, 208
- X-Content-Type-Options: nosniff* header, 203
- XDomainRequest* API, 239–240, 253
- X-Frame-Options* header, 179–180, 243

- XHTML, 12
 - and HTML entities, 78
 - minimal fault-tolerance of parser, 73
 - named entities, 76
 - syntax, 70
- XML (Extensible Markup Language)
 - and bandwidth, 123n
 - binary-only serialization, 123n
 - case of tags, 72
 - <![CDATA[...]]> blocks, 72, 78, 250
- XML Binding Language files, 90
- XML browser applications (XBAP), 135–136
- XML documents
 - browser support, 119–124
 - generic view, 120–121
- XMLHttpRequest* API, 12, 54, 210, 236, 237–238
 - httponly* cookies and, 150
 - same-origin policy mechanism for, 146–148
- xmlns* namespace, 72, 119
- <?xml-stylesheet href=... ?> directive, 88
- XML User Interface Language (XUL), 122–123
- XMLSocket, TCP connections via, 156
- <xml> tag (HTML), 72
- XSRF (cross-site request forgery), 84, 190, 262
 - exploitation of flaws, 190
 - login forms and, 145–146
- XSS (cross-site scripting), 71, 262
 - bugs, and password managers, 228
 - exploitation of flaws, 190
 - filtering, 251–252, 253
- XUL (XML User Interface Language), 122–123
- XXE (External XML Entity) attack, 76

Z

- ZIP files, extracting content from, 37
- Zone.Identifier* metadata, Internet Explorer and, 231
- zone model, for Internet Explorer, 229–231