

Inhalt

1	Sicherheit in einer Welt der Webanwendungen	1
1.1	Informationssicherheit kurz und bündig	1
1.1.1	Liebäugeln mit formalen Lösungen	2
1.1.2	Risikomanagement und IT-Sicherheit	5
1.1.3	Erleuchtung durch Taxonomie	7
1.1.4	Auf dem Weg zu praktischen Ansätzen	8
1.2	Eine kurze Geschichte des Web	9
1.2.1	Geschichten aus der Steinzeit: 1945 bis 1994	10
1.2.2	Die ersten Browserkriege: 1995 bis 1999	12
1.2.3	Das Zeitalter der Langeweile: 2000 bis 2003	14
1.2.4	Web 2.0 und die zweiten Browserkriege: 2004 und später	16
1.3	Die Entwicklung einer Bedrohung	17
1.3.1	Der Benutzer als Sicherheitslücke	17
1.3.2	Die Cloud oder die Freuden gemeinschaftlichen Lebens	18
1.3.3	Unvereinbare Vorstellungen	19
1.3.4	Browserübergreifende Interaktion: Synergie des Versagens	20
1.3.5	Die Aufhebung der Grenze zwischen Client und Server	22

Teil I: Anatomie des Web

2	Am Anfang war die URL	29
2.1	Struktur einer URL	30
2.1.1	Schema und Protokoll	30
2.1.2	Kennzeichen hierarchischer URLs	31
2.1.3	Anmeldeinformationen	32
2.1.4	Die Serveradresse	33
2.1.5	Der Serverport	34
2.1.6	Hierarchische Dateipfade	35
2.1.7	Der Query-String	35
2.1.8	Der Fragment-Identifier	36
2.1.9	Die URL im Ganzen	36

2.2	Reservierte Zeichen und URL-Codierung	39
2.2.1	Umgang mit Text jenseits der ASCII-Welt	41
2.3	Übliche URL-Schemas und ihre Funktion	45
2.3.1	Vom Browser unterstützte Protokolle für den Dokumentabruf	45
2.3.2	Protokolle für Drittanbieteranwendungen und Plug-ins	46
2.3.3	Nicht kapselnde Pseudoprotokolle	46
2.3.4	Kapselnde Pseudoprotokolle	47
2.3.5	Ein letztes Wort zur Schemaerkennung	48
2.4	Auflösung relativer URLs	48
3	HTTP	53
3.1	Die grundlegende Syntax von HTTP	54
3.1.1	Konsequenzen aus der Unterstützung von HTTP/0.9	56
3.1.2	Besonderheiten bei der Verarbeitung von Zeilenwechseln ..	57
3.1.3	Proxy-Requests	58
3.1.4	Auflösung von doppelten und widersprüchlichen Headern ..	60
3.1.5	Durch Semikolons getrennte Header-Werte	61
3.1.6	Zeichensatz und Codierungsschema für Header	62
3.1.7	Verhalten von Referer-Headern	64
3.2	HTTP-Requesttypen	65
3.2.1	GET	65
3.2.2	POST	66
3.2.3	HEAD	66
3.2.4	OPTIONS	67
3.2.5	PUT	67
3.2.6	DELETE	67
3.2.7	TRACE	67
3.2.8	CONNECT	68
3.2.9	Weitere HTTP-Methoden	68
3.3	HTTP-Statuscodes	68
3.3.1	200–299: Erfolg	68
3.3.2	300–399: Umleitung und andere Statusmeldungen	69
3.3.3	400–499: Clientseitige Fehler	70
3.3.4	500–599: Serverseitige Fehler	71
3.3.5	Konsistente Verwendung der HTTP-Codes	71
3.4	Keepalive-Sessions	71
3.5	Portionsweise Datenübertragung	73
3.6	Caching-Verhalten	74
3.7	Semantik von HTTP-Cookies	76
3.8	HTTP-Authentifizierung	79

3.9	Verschlüsselung auf Protokollebene und Clientzertifikate	80
3.9.1	Extended Validation SSL	82
3.9.2	Regeln zur Fehlerbehandlung	83
4	HTML	87
4.1	Grundprinzipien von HTML-Dokumenten	88
4.1.1	Dokumentenparser-Modi	89
4.1.2	Der Kampf um die Semantik	91
4.2	Das Verhalten von HTML-Parsern	92
4.2.1	Wechselwirkungen zwischen mehreren Tags	93
4.2.2	Explizite und implizite Bedingungen	94
4.2.3	Überlebensstrategien für die HTML-Analyse	95
4.3	Entity Encoding	95
4.4	Die Verzahnung von HTTP und HTML	97
4.5	Hyperlinks und Einbindung externer Inhalte	99
4.5.1	Einfache Links	99
4.5.2	Formulare und durch Formulare ausgelöste Requests	100
4.5.3	Frames	102
4.5.4	Typspezifisches Einbinden von Inhalten	103
4.5.5	Ein Hinweis zum Thema CSRF	105
5	CSS	109
5.1	Grundlagen der CSS-Syntax	110
5.1.1	Eigenschaften definieren	111
5.1.2	@-Direktiven und XBL-Bindings	112
5.1.3	Wechselwirkungen mit HTML	113
5.2	Risiken der Parser-Resynchronisierung	113
5.3	Zeichencodierung	114
6	JavaScript im Browser	119
6.1	Grundmerkmale von JavaScript	120
6.1.1	Die Skriptverarbeitung	121
6.1.2	Steuerung der Ausführungsreihenfolge	125
6.1.3	Möglichkeiten zum Inspizieren von Code und Objekten	126
6.1.4	Die Laufzeitumgebung anpassen	127
6.1.5	JSON und andere Arten der Datenserialisierung	129
6.1.6	E4X und andere Syntaxerweiterungen	132
6.2	Standard-Objekthierarchie	133
6.2.1	Das DOM	136
6.2.2	Zugriff auf andere Dokumente	138
6.3	Zeichencodierung im Skript	139

6.4	Möglichkeiten zum Einbinden von Code und Verschachtelungsrisiken	140
6.5	Totgesagte leben länger: Visual Basic Script	142
7	Nicht-HTML-Dokumente	145
7.1	Klartextdateien	145
7.2	Bitmap-Bilder	146
7.3	Audio und Video	147
7.4	XML-Dokumente	147
7.4.1	Allgemeine XML-Darstellung	148
7.4.2	SVG	150
7.4.3	MathML	151
7.4.4	XUL	151
7.4.5	WAP und WML	152
7.4.6	RSS- und Atom-Feeds	153
7.5	Ein Hinweis zu nicht darstellbaren Dateitypen	154
8	Inhalte mit Browser-Plug-ins darstellen	157
8.1	Plug-ins aufrufen	157
8.1.1	Die Gefahren bei der Verarbeitung von Inhaltstypen für Plug-ins	159
8.2	Helper für die Dokumentdarstellung	161
8.3	Plug-in-basierte Anwendungsframeworks	162
8.3.1	Adobe Flash	163
8.3.2	Microsoft Silverlight	166
8.3.3	Sun Java	167
8.3.4	XBAP	168
8.4	ActiveX-Steuerelemente	169
8.5	Mit anderen Plug-ins leben	170

Teil II: Sicherheitsfeatures von Browsern

9	Inhalte isolieren	177
9.1	SOP für das DOM	177
9.1.1	document.domain	180
9.1.2	postMessage(...)	181
9.1.3	Wechselwirkung mit sensiblen Browserdaten	183
9.2	SOP für XMLHttpRequest	184
9.3	SOP für Web Storage	186

9.4	Sicherheitsrichtlinie für Cookies	187
9.4.1	Der Einfluss von Cookies auf die SOP	189
9.4.2	Probleme mit Domäneneinschränkungen	190
9.4.3	Die ungewöhnliche Gefahr von »localhost«	191
9.4.4	Cookies und »legitimes« DNS-Hijacking	193
9.5	Sicherheitsregeln für Plug-ins	193
9.5.1	Adobe Flash	194
9.5.2	Microsoft Silverlight	198
9.5.3	Java	199
9.6	Umgang mit unklaren oder unerwarteten Ursprungsangaben	200
9.6.1	IP-Adressen	200
9.6.2	Hostnamen mit zusätzlichen Punkten	201
9.6.3	Nicht vollständig qualifizierte Hostnamen	201
9.6.4	Lokale Dateien	202
9.6.5	Pseudo-URLs	203
9.6.6	Browsererweiterungen und Benutzerschnittstelle	204
9.7	Andere Verwendungen für Origin-Angaben	204
10	Ursprungsvererbung	207
10.1	Ursprungsvererbung für about:blank	207
10.2	Ursprungsvererbung für data:-URLs	210
10.3	Ursprungsvererbung für javascript:- und vbscript:-URLs	212
10.4	Ein Hinweis zu eingeschränkten Pseudo-URLs	214
11	Die Welt außerhalb von SOPs	217
11.1	Fenster- und Frame-Interaktionen	218
11.1.1	Navigationsziele vorhandener Dokumente ändern	218
11.1.2	Unerwünschtes Framing	224
11.2	Domänenübergreifendes Einbinden von Inhalten	227
11.2.1	Ein Hinweis zu ursprungsübergreifenden Unterressourcen	229
11.3	Datenschutzrelevante Seitenkanäle	230
11.4	Weitere SOP-Lücken und ihre Anwendungsfälle	232
12	Sonstige Schlupflöcher	235
12.1	Navigation in sensiblen Schemas	235
12.2	Zugriff auf interne Netzwerke	237
12.3	Verbotene Ports	239
12.4	Einschränkungen für Third-Party-Cookies	241

13	Mechanismen zur Inhaltserkennung	245
13.1	Dokumenttypen erkennen	246
13.1.1	Nicht wohlgeformte MIME-Typen	247
13.1.2	Besondere Werte für den Inhaltstyp	248
13.1.3	Nicht erkannte Inhaltstypen	250
13.1.4	Defensive Verwendung des Content-Disposition-Headers .	252
13.1.5	Inhaltsdirektiven für Unterressourcen	254
13.1.6	Downloads und andere Nicht-HTTP-Inhalte	255
13.2	Zeichensätze erkennen	256
13.2.1	BOM: Kennzeichnung der Bytereihenfolge	259
13.2.2	Vererbung und Überschreiben von Zeichensätzen	260
13.2.3	Markup-gesteuerte Zeichensätze für Unterressourcen	261
13.2.4	Nicht-HTTP-Dateien erkennen	263
14	Umgang mit schädlichen Skripten	267
14.1	Denial-of-Service-Angriffe	268
14.1.1	Einschränkungen für Ausführungszeit und Speichernutzung	269
14.1.2	Verbindungseinschränkungen	270
14.1.3	Pop-up-Filter	272
14.1.4	Einschränkung der Verwendung von Dialogen	273
14.2	Probleme bei Positionierung und Darstellung von Fenstern	275
14.3	Timing-Attacken auf Benutzerschnittstellen	278
15	Webseiten mit speziellen Berechtigungen	283
15.1	Vom Browser und von Plug-ins gesteuerte Zugriffsrechte	284
15.1.1	Hart codierte Domänen	285
15.2	Formulargestützte Passwortmanager	286
15.3	Das Zonenmodell des Internet Explorer	288
15.3.1	»Mark of the Web« und Zone.Identifier	291
Teil III: Ein Blick in die Zukunft		
16	Neue und zukünftige Sicherheitsfunktionen	297
16.1	Frameworks zur Erweiterung des Sicherheitsmodells	298
16.1.1	Domänenübergreifende Requests	298
16.1.2	XDomainRequest	302
16.1.3	Andere Verwendungen des Origin-Headers	303

16.2	Frameworks zur Einschränkung des Sicherheitsmodells	304
16.2.1	Content Security Policy	305
16.2.2	Sandbox-Frames	310
16.2.3	Strict Transport Security	313
16.2.4	Private Browsing	315
16.3	Andere Entwicklungen	316
16.3.1	Browserinterne HTML-Filterung	316
16.3.2	XSS-Filter	318
17	Weitere wichtige Browsermechanismen	323
17.1	Vorschläge für URLs und Protokolle	323
17.2	Funktionen auf Inhaltsebene	326
17.3	I/O-Schnittstellen	328
18	Allgemeine Schwachstellen im Web	329
18.1	Spezifische Schwachstellen von Webanwendungen	329
18.1.1	Cross-Site Request Forgery (CSRF, XSRF oder »Sea-Surf«)	329
18.1.2	Cross-Site Script Inclusion (XSSI)	330
18.1.3	Cross-Site-Scripting (XSS)	330
18.1.4	Header Injection/Response Splitting	330
18.1.5	Mixed Content	331
18.1.6	Open Redirection	331
18.1.7	Referrer-Leaks (Referrer Leaking)	331
18.2	Probleme beim Design von Webanwendungen	332
18.2.1	Cache Poisoning	332
18.2.2	Clickjacking	332
18.2.3	Content-Sniffing/Charset-Sniffing	332
18.2.4	Cookie Forcing/Cookie Injection	332
18.2.5	DoS-Angriffe (Denial of Service)	333
18.2.6	Frame Busting	333
18.2.7	HTTP-Downgrades	333
18.2.8	Network Fenceposts	333
18.3	Häufige Probleme bei serverseitigem Code	334
18.3.1	Buffer Overflow	334
18.3.2	Command-Injection (SQL, Shell, PHP usw.)	334
18.3.3	Directory/Path Traversal	335
18.3.4	File Inclusion	335
18.3.5	Format-String-Schwachstellen	335
18.3.6	Integer Overflow	336
18.3.7	Schwachstellen in der Pointer-Arithmetik	336
Epilog		337

Teil IV: Anhang

Literatur	341
Index	353