

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Über dieses Buch . . . . .	1
1.2	Was ist Sicherheit? . . . . .	3
1.3	Wichtige Begriffe . . . . .	5
1.4	Sicherheitskonzepte . . . . .	7
1.5	ISO 17799 . . . . .	9
1.6	Wie verkaufe ich Sicherheit? . . . . .	10
1.7	Wichtige Informationsquellen . . . . .	12
	1.7.1 Mailinglisten . . . . .	12
	1.7.2 Full Disclosure . . . . .	13
	1.7.3 BugTraq . . . . .	14
	1.7.4 Webappsec . . . . .	14
1.8	OWASP . . . . .	15
1.9	PHP-Sicherheit.de . . . . .	16
<b>2</b>	<b>Informationsgewinnung</b>	<b>17</b>
2.1	Grundlagen . . . . .	17
2.2	Webserver erkennen . . . . .	18
	2.2.1 Server-Banner erfragen . . . . .	19
	2.2.2 Webserver-Verhalten interpretieren . . . . .	21
	2.2.3 Tools für Webserver-Fingerprinting . . . . .	22
2.3	Betriebssystem erkennen . . . . .	22
2.4	PHP-Installation erkennen . . . . .	23
2.5	Datenbanksystem erkennen . . . . .	25

---

2.6	Datei-Altlasten .....	27
2.6.1	Temporäre Dateien .....	27
2.6.2	Include- und Backup-Dateien .....	28
2.6.3	Dateien von Entwicklungswerkzeugen .....	29
2.6.4	Vergessene oder »versteckte« PHP-Dateien .....	29
2.7	Pfade .....	30
2.7.1	mod_speling .....	30
2.7.2	robots.txt .....	32
2.7.3	Standardpfade .....	32
2.7.4	Pfade verkürzen .....	35
2.8	Kommentare aus HTML-Dateien .....	35
2.9	Applikationen erkennen .....	36
2.9.1	Das Aussehen/Layout .....	36
2.9.2	Das Vorhandensein bestimmter Dateien .....	36
2.9.3	Header-Felder .....	37
2.9.4	Bestimmte Pfade .....	37
2.9.5	Kommentare im Quellcode .....	37
2.10	Default-User .....	38
2.11	Google Hacking .....	39
2.12	Fazit .....	39
<b>3</b>	<b>Parametermanipulation</b>	<b>41</b>
3.1	Grundlagen .....	41
3.2	Werkzeuge zur Parametermanipulation .....	44
3.2.1	Parametermanipulation mit dem Browser .....	44
3.2.2	Einen Proxy benutzen .....	47
3.3	Angriffsszenarien und Lösungen .....	49
3.3.1	Fehlererzeugung .....	49
3.3.2	HTTP Response Splitting .....	51
3.3.3	Remote Command Execution .....	55
3.3.4	Angriffe auf Dateisystemfunktionen .....	57
3.3.5	Angriffe auf Shell-Ebene .....	58
3.3.6	Cookie Poisoning .....	59
3.3.7	Manipulation von Formulardaten .....	60
3.3.8	Vordefinierte PHP-Variablen manipulieren .....	61
3.3.9	Spam über Mailformulare .....	61

3.4	Variablen richtig prüfen	63
3.4.1	Auf Datentyp prüfen	64
3.4.2	Datenlänge prüfen	65
3.4.3	Inhalte prüfen	66
3.4.4	Whitelist-Prüfungen	68
3.4.5	Blacklist-Prüfung	69
3.4.6	Clientseitige Validierung	70
3.5	register_globals	71
3.6	Fazit	74
<b>4</b>	<b>Cross-Site Scripting</b>	<b>75</b>
4.1	Grenzenlose Angriffe	75
4.2	Was ist Cross-Site Scripting?	76
4.3	Warum XSS gefährlich ist	77
4.4	Erhöhte Gefahr dank Browserkomfort	78
4.5	Formularvervollständigung verhindern	79
4.6	XSS in LANs und WANs	80
4.7	XSS – einige Beispiele	81
4.8	Ein klassisches XSS	82
4.9	Angriffspunkte für XSS	84
4.10	Angriffe verschleiern – XSS Cheat Sheet	85
4.11	Einfache Gegenmaßnahmen	88
4.12	XSS verbieten, HTML erlauben – wie?	90
4.12.1	BBCode	91
4.12.2	HTML-Filter mit XSS-Blacklist	92
4.12.3	Whitelist-Filtern mit »HTML Purifier«	94
4.13	Die Zwischenablage per XSS auslesen	97
4.14	XSS-Angriffe über DOM	98
4.15	XSS in HTTP-Headern	101
4.15.1	Angriffe der ersten Ordnung mit Headern	101
4.15.2	Second Order XSS per Header	101
4.16	Attack API	104
4.17	Second Order XSS per RSS	104

4.18	Cross-Site Request Forgery (CSRF) .....	105
4.18.1	CSRF als Firewall-Brecher .....	108
4.18.2	CSRF in BBCode .....	108
4.18.3	Ein erster Schutz gegen CSRF .....	109
4.18.4	CSRF-Schutzmechanismen .....	111
4.18.5	Formular-Token gegen CSRF .....	112
4.18.6	Unheilige Allianz – CSRF und XSS .....	113
<b>5</b>	<b>SQL-Injection</b> .....	<b>115</b>
5.1	Grundlagen .....	115
5.2	Auffinden von SQL-Injection-Möglichkeiten .....	117
5.2.1	GET-Parameter .....	118
5.2.2	POST-Parameter .....	119
5.2.3	Cookie-Parameter .....	120
5.2.4	Server-Variablen .....	121
5.3	Syntax einer SQL-Injection .....	124
5.3.1	Sonderzeichen in SQL .....	124
5.3.2	Schlüsselwörter in SQL .....	125
5.3.3	Einfache SQL-Injection .....	125
5.3.4	UNION-Injections .....	127
5.4	Advanced SQL-Injection .....	130
5.4.1	LOAD_FILE .....	130
5.4.2	Denial-of-Service mit SQL-Injection .....	131
5.4.3	ORDER BY Injection .....	131
5.5	Wie kann man sich vor SQL-Injection schützen? .....	133
5.5.1	Sonderzeichen maskieren .....	133
5.5.2	Ist Schlüsselwort-Filterung ein wirksamer Schutz? .....	133
5.5.3	Parameter Binding/Prepared Statements .....	134
5.5.4	Stored Procedures .....	135
5.6	Fazit .....	136
<b>6</b>	<b>Autorisierung und Authentisierung</b> .....	<b>137</b>
6.1	Beliebte Fehler in Login-Formularen .....	137
6.1.1	Falsche Request-Methode .....	137
6.1.2	Falsche SQL-Abfrage .....	138
6.1.3	SQL-Injection .....	139
6.1.4	XSS .....	140

6.2	Authentisierungssicherheit .....	140
6.2.1	SSL .....	141
6.2.2	Behandlung von Passwörtern .....	142
6.2.3	Benutzernamen und Kennungen .....	144
6.2.4	Sichere Passwörter .....	144
6.2.5	Passwort-Sicherheit bestimmen .....	148
6.2.6	Vergessene Passwörter .....	150
6.3	Spam-Vermeidung mit CAPTCHAs .....	155
<b>7</b>	<b>Sessions</b>	<b>161</b>
7.1	Grundlagen .....	161
7.2	Permissive oder strikte Session-Systeme .....	163
7.3	Session-Speicherung .....	164
7.4	Schwache Session-ID-Generierungsalgorithmen .....	166
7.5	Session-Timeout .....	167
7.6	Bruteforcing von Sessions .....	168
7.7	Session Hijacking .....	169
7.8	Session Fixation .....	171
7.9	Zusätzliche Abwehrmethoden .....	172
7.9.1	Page Ticket System .....	172
7.9.2	Session-Dateien mittels Cronjob löschen .....	173
7.9.3	Session-ID aus dem Referrer löschen .....	173
7.10	Fazit .....	174
<b>8</b>	<b>Upload-Formulare</b>	<b>175</b>
8.1	Grundlagen .....	175
8.2	Aufbau eines Upload-Formulars .....	175
8.3	PHP-interne Verarbeitung .....	176
8.4	Speicherung der hochgeladenen Dateien .....	177
8.5	Bildüberprüfung .....	178
8.6	PHP-Code in ein Bild einfügen .....	179
8.7	Andere Dateitypen überprüfen .....	180
8.8	Gefährliche Zip-Archive .....	181
8.9	Fazit .....	181

<b>9</b>	<b>Variablenfilter mit ext/filter</b>	<b>183</b>
9.1	Überblick .....	183
9.2	Installation .....	184
9.3	Die Filter-API .....	184
9.4	Verfügbare Filter .....	186
9.4.1	Validierende Filter .....	186
9.4.2	Reinigende Filter .....	187
9.5	Zahlen prüfen und filtern .....	187
9.6	Boolesche Werte .....	189
9.7	URLs validieren .....	189
9.8	IP-Adressen prüfen .....	190
9.9	Syntaxcheck für E-Mail-Adressen .....	192
9.10	Reinigende Filter .....	192
9.11	Prüfung externer Daten .....	193
9.12	Callback-Funktionen .....	194
9.13	Fazit .....	195
<b>10</b>	<b>PHP intern</b>	<b>197</b>
10.1	Fehler in PHP .....	197
10.1.1	File-Upload-Bug .....	197
10.1.2	Unsichere (De-)Serialisierung .....	198
10.1.3	Gefährliches Speicherlimit .....	198
10.1.4	Speicherproblem dank htmlentities .....	198
10.1.5	Bewertung .....	199
10.2	Bestandteile eines sicheren Servers .....	199
10.3	Unix oder Windows? .....	200
10.4	Bleiben Sie aktuell! .....	201
10.5	Installation .....	201
10.5.1	Installation als Apache-Modul .....	202
10.5.2	CGI .....	203
10.6	suExec .....	205
10.7	Safe Mode .....	207
10.7.1	Einrichtung des Safe Mode .....	208
10.7.2	safe_mode_exec_dir .....	208
10.7.3	safe_mode_include_dir .....	209
10.7.4	Umgebungsvariablen im Safe Mode .....	209
10.7.5	Safe Mode considered harmful? .....	210

10.8	Weitere PHP-Einstellungen	212
10.8.1	open_basedir	212
10.8.2	disable_functions	213
10.8.3	disable_classes	213
10.8.4	max_execution_time	214
10.8.5	max_input_time	214
10.8.6	memory_limit	214
10.8.7	Upload-Einstellungen	215
10.8.8	allow_url_fopen	216
10.8.9	allow_url_include	216
10.8.10	register_globals	217
10.9	Code-Sandboxing mit runkit	217
10.10	Externe Ansätze	219
10.10.1	suPHP	219
10.10.2	FastCGI	223
10.10.3	Das Apache-Modul mod_suid	225
10.11	Rootjail-Lösungen	229
10.11.1	BSD-Rootjails	229
10.11.2	User Mode Linux	229
10.11.3	mod_security	230
10.11.4	mod_chroot	230
10.12	Fazit	231
<b>11</b>	<b>PHP-Hardening</b>	<b>233</b>
11.1	Warum PHP härten?	233
11.1.1	Buffer Overflows	234
11.1.2	Schutz vor Pufferüberläufen im Suhosin-Patch	235
11.1.3	Schutz vor Format-String-Schwachstellen	236
11.1.4	Simulationsmodus	237
11.1.5	Include-Schutz gegen Remote-Includes und Nullbytes	237
11.1.6	Funktions- und Evaluationsbeschränkungen	239
11.1.7	Schutz gegen Response Splitting und Mailheader Injection	240
11.1.8	Variablenschutz	240
11.1.9	SQL Intrusion Detection	240
11.1.10	Logging	240
11.1.11	Transparente Cookie- und Session-Verschlüsselung	241
11.1.12	Härtung des Speicherlimits	241
11.1.13	Kryptographische Funktionen	241

11.2	Prinzipien hinter Suhosin .....	242
11.3	Installation .....	242
11.3.1	Installation des Patches .....	243
11.3.2	Installation der Extension .....	245
11.4	Zusammenarbeit mit anderen Zend-Extensions .....	246
11.5	Konfiguration .....	247
11.5.1	Generelle Optionen .....	247
11.5.2	Log-Dateien .....	250
11.5.3	Alarm-Skript .....	253
11.5.4	Transparente Verschlüsselung .....	254
11.5.5	Variablenfilter .....	255
11.5.6	Upload-Konfiguration .....	257
11.6	Beispielkonfiguration .....	259
11.7	Fazit und Ausblick .....	260
<b>12</b>	<b>Webserver-Filter für Apache</b> .....	<b>261</b>
12.1	Einsatzgebiet von Filtermodulen .....	261
12.2	Blacklist oder Whitelist? .....	262
12.3	mod_security .....	263
12.3.1	So funktioniert's .....	264
12.3.2	Gefahren durch mod_security .....	264
12.3.3	Installation .....	265
12.3.4	Konfiguration .....	266
12.3.5	Regelwerk von mod_security .....	269
12.3.6	Alarm-Skript für mod_security .....	279
12.3.7	Rootjail-Umgebungen mit mod_security .....	279
12.3.8	mod_security 2.0 .....	281
12.4	mod_parmguard .....	284
12.4.1	So funktioniert's .....	285
12.4.2	Installation .....	285
12.4.3	Webserver-Konfiguration .....	287
12.4.4	XML-Whitelist manuell erstellen .....	288
12.4.5	Automatische Erzeugung .....	293
12.5	Fazit .....	294

---

<b>Anhang</b>	<b>295</b>	
<b>A</b>	<b>Checkliste für sichere Webapplikationen</b>	<b>297</b>
<b>B</b>	<b>Wichtige Optionen in php.ini</b>	<b>301</b>
B.1	variables_order . . . . .	301
B.2	register_globals . . . . .	302
B.3	register_long_arrays . . . . .	302
B.4	register_argc_argv . . . . .	302
B.5	post_max_size . . . . .	303
B.6	magic_quotes_gpc . . . . .	303
B.7	magic_quotes_runtime . . . . .	303
B.8	always_populate_raw_post_data . . . . .	303
B.9	allow_url_fopen . . . . .	304
B.10	allow_url_include . . . . .	304
<b>C</b>	<b>Liste aller Schwachstellen mit Gefahrenpotenzial-Bewertung</b>	<b>305</b>
C.1	Cross-Site Scripting . . . . .	305
C.2	Information Disclosure . . . . .	305
C.3	Full Path Disclosure . . . . .	306
C.4	SQL-Injection . . . . .	306
C.5	HTTP Response Splitting . . . . .	306
C.6	Cross-Site Request Forgery . . . . .	307
C.7	Remote Command Execution . . . . .	307
C.8	Mail-Header Injection . . . . .	307
<b>D</b>	<b>Glossar</b>	<b>309</b>
	<b>Stichwortverzeichnis</b>	<b>317</b>