

Dominik Schadow

Java-Web-Security

Sichere Webanwendungen mit Java entwickeln



dpunkt.verlag

Dominik Schadow
info@dominikshadow.de

Lektorat: René Schönfeldt
Copy-Editing: Friederike Daenecke, Zülpich
Satz: Da-TeX, Leipzig
Herstellung: Frank Heidt
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:
Buch 978-3-86490-146-1
PDF 978-3-86491-448-5
ePub 978-3-86491-449-2

1. Auflage
Copyright © 2014 dpunkt.verlag GmbH
Wieblinger Weg 17
691123 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Über dieses Buch	2
1.2	Zielgruppe und Voraussetzungen	3
1.3	Webanwendungen	4
1.4	Abgrenzung	5
1.5	Der Quellcode zum Buch	6
1.6	Aufbau des Buches	7
1.7	Danksagungen	9
2	Sicherheit von Anfang an	11
2.1	Forderung nach Sicherheit	11
2.2	Warum ist sichere Software wichtig?	13
2.3	Wer muss sicher entwickeln?	14
2.4	Sicherheit in allen Phasen	16
2.5	Veränderungen im Entwicklungsprozess	18
	2.5.1 Klärung der notwendigen Sicherheitsanforderungen .	19
	2.5.2 Risikoanalyse	19
	2.5.3 Sicherheit einplanen	21
	2.5.4 Code-Reviews	22
	2.5.5 Ganzheitliche Sicherheit	22
2.6	Der Preis der Sicherheit	24
2.7	Sichere Webapplikationen entwickeln	25
	2.7.1 Altapplikationen absichern	26
	2.7.2 Web Application Firewalls	27
2.8	Absolute Sicherheit gibt es nicht	29
2.9	Auf einen Blick	29
3	Java ist doch schon sicher?!	31
3.1	Grundlagen	31
3.2	Java-Features rund um die Sicherheit	33
3.3	Was Java nicht leisten kann	34
3.4	Welche Java-Versionen sind betroffen?	35
3.5	Sichere Entwicklung mit Java	36

3.5.1	Open Web Application Security Project	38
3.5.2	CWE/SANS	40
3.6	Auf einen Blick	42
4	Java-Security-Basics	43
4.1	Security-Frameworks	43
4.1.1	Enterprise Security API	47
4.1.2	Coverity Security Library	47
4.1.3	Korrekte Verwendung	48
4.2	Input-Validierung	49
4.2.1	Threat Modeling	51
4.2.2	Validierungsregeln	52
4.2.3	Validierung aller Benutzereingaben	55
4.2.4	Validierung in Frontend und Backend	59
4.2.5	Frameworks	61
4.3	Output-Escaping	62
4.3.1	Grundlagen	63
4.3.2	Frameworks	67
4.4	Fehlerbehandlung	70
4.5	Auf einen Blick	72
4.5.1	Beispielprojekte	73
4.5.2	Checkliste	73
5	Session-Management mit Java	75
5.1	Grundlagen	75
5.1.1	Frühzeitige Klärung der Anforderungen	76
5.1.2	Transportsicherheit	77
5.2	Session-Handling und Session-ID	79
5.2.1	Session-Fixation	81
5.2.2	HTTP Strict Transport Security	83
5.2.3	Cookies	88
5.2.4	Sessiondaten im Cookie speichern	89
5.2.5	Vollständige Konfiguration der web.xml	91
5.3	Authentifizierung und Autorisierung	91
5.3.1	Presentation Layer Access Control	92
5.3.2	Anwendungen für Benutzer und Administratoren	100
5.4	Verwendung von Frameworks	101
5.5	Auf einen Blick	102
5.5.1	Beispielprojekte	102
5.5.2	Checkliste	103

6	Injections	105
6.1	Grundlagen	105
6.2	SQL Injection	107
6.2.1	Was kann passieren?	109
6.2.2	Wie läuft ein Angriff ab?	110
6.2.3	Was können Sie dagegen tun?	114
6.3	Weitere Injections	124
6.3.1	XPath Injection	125
6.3.2	Log Injection	129
6.4	Auf einen Blick	131
6.4.1	Beispielprojekte	131
6.4.2	Checkliste	132
7	Cross-Site Scripting (XSS)	133
7.1	Grundlagen	133
7.2	Was kann passieren?	137
7.3	Wie läuft ein Angriff ab?	139
7.3.1	Stored XSS	142
7.3.2	Reflected XSS	144
7.3.3	DOM Based XSS	146
7.4	Was können Sie dagegen tun?	148
7.4.1	Session-Informationen schützen	149
7.4.2	Input-Validierung	153
7.4.3	Output-Escaping	155
7.4.4	Content Security Policy (CSP)	164
7.4.5	Browsererkennung von XSS	170
7.5	Auf einen Blick	172
7.5.1	Beispielprojekte	173
7.5.2	Checkliste	174
8	Cross-Site Request Forgery (CSRF)	175
8.1	Grundlagen	175
8.2	Was kann passieren?	180
8.3	Wie läuft ein Angriff ab?	182
8.4	Was können Sie dagegen tun?	183
8.4.1	Begrenzung der Sessiondauer	184
8.4.2	Formulare per HTTP POST übertragen	186
8.4.3	Captchas	189
8.4.4	Verwendung eines Anti-CSRF-Tokens	190
8.5	Kombination von CSRF- und XSS-Angriffen	202
8.6	Auf einen Blick	203
8.6.1	Beispielprojekte	204
8.6.2	Checkliste	204

9	Tools	205
9.1	Codeanalyse und Codequalität	205
9.1.1	Überblick	206
9.1.2	FindBugs	208
9.1.3	PMD	210
9.1.4	OWASP Dependency Check	210
9.1.5	Weitere Tools	212
9.2	Analyse und Training	212
9.2.1	Überblick	214
9.2.2	OWASP ZAP	214
9.2.3	OWASP Security Shepherd	217
9.2.4	OWASP Broken Web Applications Project	218
9.2.5	Weitere Tools	219
9.3	Auf einen Blick	220
9.3.1	Checkliste	220
10	Ausblick	221
10.1	Was Sie jetzt beherrschen	221
10.2	Weitere Themen	222
10.3	Nächste Schritte	223
10.3.1	Security Testing	223
10.3.2	Security Reviews	224
10.3.3	Security Development Lifecycle	224
10.4	Fazit	225
	Anhänge	227
A	CSRF und Webservices	229
B	Weitere Security-Frameworks	231
B.1	Spring Security	231
B.2	Apache Shiro	232
C	Abkürzungen	235
	Literatur – offline und online	237