



Xpert.press

Walter Kriha
Roland Schmitz

Sichere Systeme

 Springer

Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals in den Bereichen Softwareentwicklung, Internettechnologie und IT-Management aktuell und kompetent relevantes Fachwissen über Technologien und Produkte zur Entwicklung und Anwendung moderner Informationstechnologien.

Walter Kriha · Roland Schmitz

Sichere Systeme

Konzepte, Architekturen und Frameworks

 Springer

Prof. Dr. Walter Kriha
Hochschule der Medien
Nobelstraße 10
70569 Stuttgart
kriha@hdm-stuttgart.de

Prof. Dr. Roland Schmitz
Hochschule der Medien
Nobelstraße 10
70569 Stuttgart
schmitz@hdm-stuttgart.de

ISBN 978-3-540-78958-1

e-ISBN 978-3-540-78959-8

DOI 10.1007/978-3-540-78959-8

Xpert.press ISSN 1439-5428

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.

© 2009 Springer-Verlag Berlin Heidelberg

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Einbandgestaltung: KünkelLopka, Heidelberg

Gedruckt auf säurefreiem Papier

9 8 7 6 5 4 3 2 1

springer.de

Inhaltsverzeichnis

1	Einführung und Motivation	1
	Literatur	9
2	Angriffe	11
2.1	Eine Übersicht der Attacken	12
2.1.1	Wie lassen sich die Angriffe klassifizieren?	12
2.1.2	Eine Übersicht der häufigsten Attacken auf Applikationen	13
2.1.3	Lokale Angriffsformen	15
2.1.4	Zeitliche Entwicklung der Attacken	15
2.2	Die Attacken im Einzelnen	18
2.2.1	Externe Attacken auf Applikationslevel	18
2.2.2	Lokale Attacken auf Applikationslevel	30
2.3	Grundlagen der Input-Validierung	38
2.3.1	Abwehr auf der Netzwerkschicht	39
2.3.2	Input-/Output-Validierung auf Applikationsebene	48
2.3.3	Forms-Validierung	50
2.4	Attacken auf Ebene der Semantik	61
2.4.1	Die Kunst der Täuschung	61
2.4.2	Phishing und Multi-Faktor-Authentisierung	63
2.4.3	Soziale Attacken	67
2.5	WEB2.0-Techniken und ihre Problematik	69
2.5.1	Attacken auf Web2.0	70
2.5.2	Die Techniken hinter Web2.0	72
2.5.3	Spezielle Aspekte der AJAX-Security	77
2.6	Zur Psychologie der Verteidigung	84
	Literatur	88

3	Grundprinzipien des Designs sicherer Systeme	91
3.1	Economy of Mechanism (KISS).....	93
3.2	Fail-Safe-Defaults.....	94
3.3	Complete Mediation	94
3.4	Open Design: The Design Should Not Be Secret	97
3.5	Separation of Privilege	98
3.6	Principle of Least Privilege (POLP).....	99
3.7	Least Common Mechanism	100
3.8	Psychological Acceptability	103
3.9	Zusammenfassung der Prinzipien von Saltzer und Schroeder.....	104
3.10	Principle of Least Authority – POLA	105
3.10.1	POLA vs. POLP.....	105
3.10.2	Access Control Lists und Sandboxes	108
3.10.3	Capabilities	111
3.10.4	Zusammenfassung	113
	Literatur	114
4	Platform Security	115
4.1	Platform Security heute	116
4.2	Die „Immutable Laws of Security“: Schadensbegrenzung unmöglich?.....	121
4.3	Sicherheit und Zuverlässigkeit von Betriebssystemen	126
4.3.1	Klassische Sicherheitsarchitektur in Betriebssystemen	127
4.3.2	Kernel-Architektur: Microkernel vs. monolithischer Kernel.....	130
4.3.3	Die Sicherheit von Plug-In-Architekturen	134
4.4	Kapselung durch virtuelle Maschinen	139
4.4.1	Sicherheitsumfeld	139
4.4.2	Paravirtualisierte VM-Architekturen	141
4.4.3	Hybride Systeme im Embedded-Control-Bereich.....	142
4.4.4	Virtualisierung unter Einschluss der Verfügbarkeit: Mainframes	143
4.4.5	Grenzen der Sicherheit durch Virtualisierung.....	144
4.5	Softwarebasierte Isolation am Beispiel Singularity	145
4.6	Das Least Authority (POLA) Prinzip beim Bau von Systemen	151
4.6.1	Additive und subtraktive Rechtezuweisung.....	152
4.6.2	Labeling und Tainting.....	153
4.6.3	Authority Reduction in Vista	154
4.6.4	Privilegienanpassung in Betriebssystemen	157
4.6.5	Privilegienanpassung in Applikationen.....	159

- 4.7 Die Sicherheit von Servern..... 160
 - 4.7.1 Grundkonzepte der Serversicherheit..... 161
 - 4.7.2 Sicherheitsrelevante Architekturmerkmale von Servern..... 165
 - 4.7.3 Absicherung von Servern am Beispiel des OKWS-Web-Servers 169
- 4.8 Administration..... 174
 - 4.8.1 Race Conditions..... 175
 - 4.8.2 SU – verschiedene Implementationen..... 175
 - 4.8.3 Installation 179
- Literatur 180

- 5 Java Sprach- und Plattformsicherheit 183**
 - 5.1 Die Evolution der Java-Sicherheit..... 184
 - 5.2 Sprachbasierte Sicherheit 187
 - 5.2.1 Memory Protection und Typsicherheit..... 187
 - 5.2.2 Erweiterungen und Ableitungen von Klassen..... 193
 - 5.3 Access Control Policies und ihre Implementation..... 195
 - 5.3.1 Grundlegende Aspekte von Policies 195
 - 5.3.2 Java Code Access Security 198
 - 5.3.3 Codebasierte Policies..... 203
 - 5.3.4 Userbasierte Policies mit JAAS 207
 - 5.3.5 Alternative Methoden der Zugriffskontrolle..... 217
 - 5.4 Designaspekte der Umstellung auf Java 2 Security..... 218
 - 5.5 Code Security für Enterprise-Applikationen 223
 - Literatur 226

- 6 Enterprise Security 229**
 - 6.1 Grundlagen von Enterprise Security..... 230
 - 6.1.1 Vom Application Tower zur verteilten Enterprise-Applikation..... 230
 - 6.1.2 Security-Infrastruktur..... 232
 - 6.1.3 Serviceorientierte Architekturen (SOA) 234
 - 6.1.4 Eventgetriebene Systeme und Busse..... 235
 - 6.2 Problematik der Ende-zu-Ende-Sicherheit 239
 - 6.2.1 Speichern von Passwörtern in Registries 240
 - 6.2.2 Nicht-authentisierter Zugriff durch den Suchdienst..... 241
 - 6.2.3 Zugriff durch Patch-Services 244
 - 6.2.4 Propagation von Identität, Recht und Aufruf..... 245
 - 6.3 JAVA EE und EJB 248
 - 6.3.1 Grundprinzipien sicherer Software in JAVA EE und EJB..... 250
 - 6.3.2 JAVA-EE-Applikationsinterfaces und ihre Verwendung..... 251
 - 6.3.3 Rolle, Permission und Ressource..... 253

6.3.4	Delegation und Impersonation	256
6.3.5	Rollenbasierte, deklarative Access Control anhand eines Beispiels	262
6.3.6	Deklarative und programmatische Zugriffskontrolle	272
6.3.7	Annotationen für Sicherheit in Java EE	276
6.3.8	Sicherheitsdiskussion der JAVA-EE-Sicherheit	277
6.4	Struts	278
6.4.1	Authentisierung in Web-Applikationen	278
6.4.2	Spezielle Probleme der Front-End-Applikationen	288
6.4.3	Authentisierungsobjekte	289
6.4.4	Autorisierung	290
6.5	Light-weight Container Security: Das Spring Security (Acegi) Framework	295
6.5.1	Philosophie	295
6.5.2	Grundprinzipien	296
6.5.3	Authentisierung	297
6.5.4	Das Authentication-Objekt	298
6.5.5	Authorisierung	299
6.5.6	Diskussion der Spring Security Security	302
6.6	Grids	303
6.6.1	Grid Security versus Enterprise Security	303
6.6.2	Grid Security Standards	305
6.6.3	Qualifizierte Delegation durch Proxy-Zertifikate	306
6.7	Integration von Conventional-off-the-Shelf (COTS) Applikationen	309
	Literatur	310
7	Application Server Security	313
7.1	Vom Socket-Server zur Container-Architektur	314
7.1.1	Authentisierungs-Frameworks	315
7.1.2	Concurrency und Multithreading	316
7.1.3	Komponentenarchitektur	316
7.1.4	Vom Standalone-Server zum Teil eines verteilten Systems	319
7.1.5	Server-Cluster	319
7.1.6	Anforderungen an die Entwickler	320
7.2	Beispielarchitektur eines Applikationsservers	320
7.2.1	System Context und Security Context	321
7.2.2	Verwaltung von Credentials	322
7.2.3	Formen der Authentisierung	324
7.2.4	Der Classloader	325
7.3	Security-Interfaces zwischen Server und Applikationen	328
7.4	Repräsentation von Identität: Die Erstellung eines Subjects	331
7.4.1	Grundlegende Szenarien	331
7.4.2	Initiale Authentisierung	332

7.4.3	Bereits diesem Application-Server bekannter User	333
7.4.4	Bereits gegenüber einem Application-Server bekannter User	334
7.4.5	Identity Assertion durch vorgelagerte Authentisierung	337
7.4.6	Identity Assertion und Delegation	339
7.4.7	Standard-Interface für Authorization-Provider: JACC ...	341
7.4.8	Erzeugung von Subjects im Application-Server-Verbund	342
7.5	Probleme zwischen JAAS und JAVA EE.....	344
7.6	Absichern von Application-Servern („Hardening“)	347
7.6.1	Auslieferungszustand vs. Default-Is-Deny-Prinzip.....	348
7.6.2	Absicherung der Verbindungen	349
7.6.3	Absicherung der Ressourcen.....	350
7.6.4	Verwaltung des Runtime-Systems	350
7.6.5	Isolation	351
7.6.6	Cross-Cell-Trust.....	353
7.6.7	Code Access Control in Application-Servern	354
7.6.8	„Fully Trusted“ .NET.....	355
7.6.9	Isolierende VMs und Application-Server: SAP	356
7.6.10	User Identity Propagation zu Datenbanken.....	358
7.7	Zusammenfassung	361
	Literatur	362
8	Sichere Multi-Vendor Komponentensysteme.....	363
8.1	Dynamische Services und ihre Isolation in OSGI	364
8.1.1	Anwendungsbeispiele in Heimnetzwerken	365
8.1.2	Namespaces durch Classloader-Architektur	366
8.1.3	Trusted Code und Rechtezuweisung	369
8.2	Sichere Transaktionen mit FINREAD	370
8.2.1	Benutzung	371
8.2.2	Bedrohungsmodell	372
8.2.3	FINREAD-Architektur	374
8.2.4	Trust- und Key-Management	376
8.2.5	Softwareumgebung zur Isolation	380
	Literatur	383
9	Web-Services und objektbasierte Sicherheit.....	385
9.1	Modelle sicherer Kollaboration	386
9.1.1	Fallstudie: E-Voting.....	388
9.1.2	Grenzen der kanalbasierten Sicherheit.....	390
9.2	Objektbasierte Sicherheit.....	390
9.2.1	XML-Signaturen.....	391
9.2.2	XML-Verschlüsselung.....	392
9.2.3	Interoperabilität.....	394

9.3	SOAP Security.....	396
9.3.1	WS-Security.....	396
9.3.2	Security Token Issuer (STI).....	398
9.3.3	WS-Trust-Szenarien.....	401
9.3.4	Autoritäten und Aussagen.....	403
9.3.5	Bedrohungen und Risiken.....	406
9.3.6	Einbindung in Applikationen.....	407
9.3.7	Web-Services über REST	407
9.4	SOA-Security	411
	Literatur	413
10	Sichere Software: Mechanismen und Konstruktionsprinzipien	415
10.1	Rahmenbedingungen sicherer Software	417
10.1.1	Die Macht des Faktischen	417
10.1.2	Die Frage der Verantwortung	418
10.2	Grundlegende Mechanismen sicherer Software	419
10.2.1	Filter.....	419
10.2.2	Isolation	420
10.2.3	Hierarchische Modi.....	421
10.2.4	Identitäts- vs. autoritätsbasierte Zugriffskontrolle	423
10.2.5	Kanalbasierte Sicherheit vs. objektbasierte Sicherheit....	425
10.2.6	Capabilities	426
10.2.7	Schlüssel	427
10.2.8	Zugriffskontrolle in lokalen und verteilten Systemen....	428
10.2.9	Infrastruktur vs. Application Level Security.....	429
10.2.10	Front-End vs. Back-End Security	430
10.2.11	Föderation	431
10.2.12	Basismechanismen der Zugriffskontrolle	431
10.3	Sicherheit und Softwarearchitektur	432
10.3.1	Perspektiven auf Software	432
10.3.2	Zur Bestimmung der Gefährlichkeit	433
10.3.3	Mikro-Architektur.....	434
10.3.4	Klassen, Objekte und Referenzen	436
10.3.5	Funktionaler Ansatz.....	440
10.3.6	Closures	442
10.3.7	Tainting.....	444
10.3.8	Object Capabilities.....	445
10.3.9	Sicherheitsbausteine.....	448
10.3.10	Makro-Architektur	453
10.3.11	Design Patterns für sichere Software	458
10.4	Concurrency	462
10.5	Gegenkräfte	473
10.5.1	Nachträgliches Eindämmen	474
10.5.2	Globale Navigation	474
10.5.3	Globale Namespaces.....	474

10.5.4	Alias-Definitionen	475
10.5.5	Degeneration von Architektur.....	475
10.5.6	Ambige Interfaces.....	476
10.5.7	Paarweise Methoden mit State.....	476
10.5.8	State halten.....	476
10.5.9	Überaggregation	477
10.6	Sicherheit der Applikation.....	477
10.7	Sind Sicherheit und Nebenläufigkeit „Aspekte“?.....	481
	Literatur	483
11	Fehleranalyse.....	485
11.1	Fehleranalyse als didaktisches Prinzip	485
11.2	Fehleranalyse als theoretisches Prinzip	486
11.3	Datenbasis und Hypothesenbildung	487
11.4	Die Fehler im Einzelnen.....	489
11.4.1	MFSA 2005-44: Privilege escalation via non-DOM property overrides.....	489
11.4.2	MFSA 2005-43 „Wrapped“ JavaScript: URLs bypass security checks.....	492
11.4.3	MFSA 2005-33 JavaScript „lambda“ replace exposes memory contents.....	493
11.4.4	MFSA 2005-34 PLUGINSPAGE privileged javascript execution	494
11.4.5	MFSA 2005-36: Cross-site scripting through global scope pollution.....	495
11.4.6	MFSA 2005-37: Code execution through JavaScript: Favicons.....	496
11.4.7	MFSA 2005-38 Search plug-in cross-site scripting	497
11.4.8	MFSA 2005-31/39 Arbitrary code execution from Firefox sidebar panel I + II.....	498
11.4.9	MFSA 2005-28: Unsafe /tmp/plugtmp directory exploitable to erase user’s files	499
11.4.10	MFSA 2005-25: Image drag and drop executable spoofing	500
11.4.11	MFSA 2005-24: http auth prompt tab spoofing.....	501
11.5	Klassifikation der Schwachstellen im Mozilla-Firefox-Browser... ..	502
11.5.1	Übersicht.....	503
11.5.2	Fehlermuster	503
11.6	Browserarchitektur und Sicherheitskonzepte	504
11.6.1	Was ist ein Browser?	504
11.6.2	Gibt es ungefährlichen Input?	505
11.6.3	Symbolische Referenzen.....	507
11.6.4	Hobbles und Patches – JavaScript im Netscape-Browser.....	507
11.6.5	Sicherheitskonzept des Browsers.....	510

11.6.6	Data-Tainting	514
11.6.7	Eine alternative Architektur	516
	Literatur	517
12	Browsersicherheit durch Object Capabilities	519
12.1	Beschreibung des Experiments.....	520
12.2	Architektur, Design und Implementation der Lösung	522
12.2.1	Makro-Architektur	522
12.2.2	Komponenten und Applikation	523
12.2.3	Taming und Capability Architecture.....	526
12.2.4	HTML	527
12.3	Sicherheitsanalyse	528
	Literatur	531
13	Security und Usability	533
13.1	Zum Verhältnis von Usability und Security	534
13.2	Lokale vs. fremdvermittelte Sicherheit.....	536
13.2.1	Praktisches Key-Management.....	537
13.2.2	Namen und ihre eindeutige Verwendung.....	540
13.2.3	Rechte und Eigentum.....	543
13.2.4	Discretionary vs. Mandatory – über Eigentum und Verfügung	544
13.2.5	Bequemlichkeit vs. Sicherheit – der richtige Gegensatz?.....	545
13.2.6	Erkennen und Ausführen von Intentionen	547
13.2.7	Psychologische Voraussetzungen der User-Interaktion	548
13.2.8	Messen von Intention – Verständnisprüfung	554
13.2.9	Autorität aus Designation	556
13.2.10	Sichere User-Interfaces	557
13.2.11	Herstellen eines Trusted Path.....	558
13.2.12	Web2.0 und die Frage der Nutzerintention	560
	Literatur	561
14	Bestimmung der Sicherheit durch formale Ansätze	563
14.1	Zur Frage der Entscheidbarkeit von Sicherheit	564
14.1.1	Die Access Control Matrix	564
14.1.2	Capabilities modelliert durch Take/Grant-Systeme	568
14.1.3	Differenzierung des Take/Grant-Ansatzes	574
14.2	Der Scoll-Ansatz in der Sicherheitsanalyse (Fred Spiessens)	580
14.2.1	Das Confused-Deputy-Problem	580
14.2.2	Kernkonzepte der Sicherheitsanalyse	582
14.2.3	KBM: Konservative Modelle für Systeme interagierender Entitäten.....	586

- 14.2.4 Scoll: Eine Sprache, um Sicherheitsprobleme auszudrücken 592
- 14.2.5 Die Analyse des Confused Deputy mit Scollar 596
- 14.3 Prüfung Operationaler Umgebungen 602
 - 14.3.1 Modellierung und Generierung 605
 - 14.3.2 Model-Checking von Sicherheitsprotokollen 607
 - 14.3.3 Modell-getriebenes Security-Engineering 607
 - 14.3.4 Lernen und Testen 609
 - 14.3.5 Beweisbare Plattformsicherheit 609
- Literatur 613

- 15 Schlussbetrachtungen 615**
 - 15.1 Security vs. Safety 615
 - 15.2 Security vs. Usability 616
 - 15.3 Mehr Sicherheit durch Einschränkung der Nutzer? 618
 - 15.4 Prozessorientierte Security 618
 - 15.5 Modellbasierte, integrierte Security 619
 - 15.6 Zuverlässigkeit und Verfügbarkeit 620
 - Literatur 621

- Index 623**

Kapitel 1

Einführung und Motivation

Warum wird so viel unsichere Software entwickelt? In dem Buch „Internet-Security aus Software-Sicht – Grundlagen“ [KS] haben wir vor allem die Mängel in der Wahrnehmung von Sicherheitsproblemen betont und versucht, diese Wahrnehmung anhand von Fallbeispielen und Sicherheitsanalysen zu stärken. Gleichzeitig wurden dort die sicherheitstechnischen Grundbausteine, Protokolle und Dienste in einer solchen Form vorgestellt, dass sie für praktisch tätige Entwickler, Architekten und Infrastrukturspezialisten verständlich und nutzbar sind.

Mit der verbesserten Wahrnehmung von Sicherheitsproblemen und Grundkenntnissen in Kryptografie allein ist es jedoch nicht getan. Wie sichert man eigentlich konkret eine Server-Applikation für das Internet ab? Gemeint sind hier nicht mehr Infrastrukturmaßnahmen, welche die Angriffsfläche innerhalb der DMZ verkleinern, sondern die Serversoftware selbst und die Plattform, auf der sie läuft. Welche Strategien und Mechanismen müssen wir einsetzen, damit unsere Software die Businessmodelle erfüllt, sodass nur autorisierte Personen Zugriff erhalten? Das Wissen über die Strategien ist in den Köpfen erfahrener Systemarchitekten enthalten, oder in den Vorschriften zur Softwareentwicklung mancher Konzerne und nicht zuletzt in einzelnen wissenschaftlichen Papieren z. B. über den Bau sicherer Web-Server. Eine Aufgabe dieses Buches ist es, dieses Wissen zusammenzutragen, zu diskutieren, die jeweiligen Grenzen der Strategien aufzuzeigen und somit für die Entwickler nutzbar zu machen.

Heute entsteht Software meist auf der Basis von Frameworks, die bereits Sicherheitstechniken z. B. für die Autorisierung von Zugriffen eingebaut haben und deren Verständnis essenziell für den Bau sicherer Systeme ist. Dies betrifft Applikationsentwickler wie Systemingenieure. Die letzteren verwenden ebenfalls Frameworks, um Applikationen mit der jeweiligen Firmeninfrastruktur zu verbinden. Somit entstehen diffizile Abhängigkeiten zwischen Infrastruktur und Applikationen. Dieser Band hat daher als einen weiteren Schwerpunkt, die Absicherung von Enterprise-Software durch Frameworks wie z. B. EJB, JAAS etc. zu erklären.

Das Buch bleibt aber nicht bei der Betrachtung einzelner Teile von Sicherheitstechniken stehen, sondern versucht das Gesamtsystem zu betrachten. Ein Gesamt-