
Inhaltsverzeichnis

1	Einleitung	1
1.1	Zum Begriff „Webanwendung“	1
1.2	Warum sind Webanwendungen unsicher?	3
1.2.1	Anwendungslayer versus Netzwerklayer	3
1.2.2	Design-Entscheidungen des HTTP-Protokolls	4
1.2.3	Man-in-the-Middle-Proxys und Browser-Plugins	9
1.2.4	Unzureichende Validierung von Eingabedaten	11
1.2.5	Webanwendungen sind häufig Mehrbenutzersysteme	13
1.2.6	Offenlegung serverseitiger Objekte und Funktionen	13
1.2.7	Enkodierung	15
1.2.8	JavaScript	17
1.2.9	Fehlende Kontrolle der Ausführungsumgebung	18
1.2.10	Hochprivilegierter Programmcode	19
1.2.11	Unzureichende Absicherung von Backendsystemen	19
1.2.12	Unsichere 3rd-Party-Komponenten und Legacy Code	20
1.2.13	Jeder kann heute eine Webanwendung zusammenbauen	22
1.2.14	Die Agilität moderner Webentwicklung	22
1.2.15	Die Gefahr wird unterschätzt	24
1.2.16	Eingeschränkte Sichtbarkeit und Zielkonflikte	26
1.3	Was ist Webanwendungssicherheit?	28
1.3.1	Begriffsbestimmung	28
1.3.2	Nicht das gleiche wie „sichere“ Webanwendungen	30
1.3.3	Etwas anderes als Websicherheit	31
1.3.4	Das Management von Risiken	31
1.3.5	Der Schutz personenbezogener Daten	33
1.3.6	Der Umgang mit Vertrauen	34
1.3.7	Softwarequalität	35
1.3.8	Nicht durch Technik alleine zu lösen	36
1.3.9	Ein Querschnittsthema	36
1.3.10	Ein evolutionärer Prozess	38

1.4	Relevante Organisationen	39
1.5	Zusammenfassung	41
2	Bedrohungen für Webanwendungen	43
2.1	Begriffe und Konzepte	43
2.1.1	Bedrohung, Bedrohungsquelle und Gefährdung	44
2.1.2	Schwachstellen und Sicherheitslücken	45
2.1.3	Angriffe	49
2.2	Relevante Standards und Projekte	54
2.3	Manipulation der Anwendungslogik	56
2.4	Pufferüberläufe (Buffer Overflows)	58
2.5	Interpreter Injection	59
2.5.1	SQL Injection	60
2.5.2	OS Command Injection	64
2.5.3	Serverseitiges Code Injection	65
2.5.4	XML Injection	67
2.6	Clientseitige Angriffe	69
2.6.1	Hintergrund: Die Same Origin Policy (SOP)	69
2.6.2	Hintergrund: Ajax-Zugriffe	71
2.6.3	Cross-Site Scripting (XSS)	72
2.6.4	Cross-Site Request Forgery (CSRF)	80
2.6.5	Cross-Site Redirection	83
2.6.6	Sitzungsfixierung (Session Fixation)	85
2.6.7	Session Hijacking	86
2.6.8	Website Spoofing & Defacement	89
2.6.9	Clickjacking	91
2.6.10	Drive by Infection („Malwareschleudern“)	92
2.7	Angriffe auf Benutzerkonten und Privilegien	94
2.7.1	Ermitteln von Passwörtern (Brute Forcing etc.)	94
2.7.2	Passwort Recycling	96
2.7.3	Ungeschützte Ressourcen	96
2.7.4	Path Traversal	98
2.7.5	Privilegienerweiterung	99
2.7.6	Überprivilegierung	101
2.7.7	Hintertüren (Backdoors)	102
2.8	Information Disclosure	103
2.9	Unbeabsichtigte Aktionen	105
2.9.1	Race Conditions	105
2.9.2	Replay und „Verklicken“	107
2.10	Denial of Service (DoS)	108
2.11	Zusammenfassung	109

3 Technische Sicherheitsmaßnahmen	113
3.1 Begriffe und Konzepte	114
3.1.1 Sicherheitsmechanismen (Security Controls)	114
3.1.2 Sicherheitsanforderungen	116
3.1.3 Quick Wins und Best Practices	117
3.1.4 Angemessene Sicherheit	117
3.2 Relevante Standards und Projekte	120
3.2.1 Bundesdatenschutzgesetz (BDSG)	120
3.2.2 PCI-DSS	121
3.2.3 Secure Coding Guidelines	122
3.2.4 BSI Grundsatzkataloge und Studien	122
3.2.5 NIST Special Publications	123
3.2.6 OWASP Guidelines und Cheat Sheets	123
3.2.7 OWASP ESAPI	124
3.2.8 ÖNORM A7700	125
3.2.9 TSS-WEB	125
3.3 Übergreifende Sicherheitsprinzipien	125
3.3.1 Kenne deine Gegner	126
3.3.2 Berücksichtige Sicherheit im Entwurf („Secure by Design“)	128
3.3.3 Verwende einen offenen Entwurf	128
3.3.4 Verwende ein positives Sicherheitsmodell	129
3.3.5 Behebe die Ursachen (Ursachenbehebungsprinzip)	130
3.3.6 Minimiere die Angriffsfläche (Minimalprinzip)	131
3.3.7 Vermeide Risiken (Vermeidungsprinzip)	133
3.3.8 Verwende Indirektionen (Indirektionsprinzip)	133
3.3.9 Minimiere Privilegien („Least Privilege“)	134
3.3.10 Implementiere Sicherheit mehrschichtig („Defense in Depth“)	135
3.3.11 Gewährleiste einen sicheren Zustand („Fail Safe“)	137
3.3.12 Verwende sichere Standardeinstellungen („Secure Defaults“)	138
3.3.13 Vertraue niemandem (Misstrauensprinzip)	139
3.3.14 Gestalte Sicherheit konsistent	140
3.3.15 Vermeide Komplexität („Keep it Simple“)	141
3.3.16 Verwende ausgereifte Sicherheit	141
3.3.17 Gestalte Sicherheit anpassbar	142
3.3.18 Berücksichtige das Maximumprinzip	143
3.3.19 Antizipiere technologische Limitationen	143
3.3.20 Nutze Technologiebrüche	144
3.3.21 Stelle Testbarkeit sicher (Testbarkeitsprinzip)	144
3.3.22 Gestalte Sicherheit benutzerfreundlich	145
3.3.23 Trainiere deine Anwender	146
3.4 Anti-Patterns	147

3.5	Kryptographie	148
3.5.1	Zentrale kryptographische Verfahren	148
3.5.2	SSL- bzw. X.509-Zertifikate	148
3.5.3	HTTPS	151
3.5.4	Zusammenfassung	154
3.6	Datenvalidierung	155
3.6.1	Das Prinzip Datenvalidierung	156
3.6.2	Eingabevalidierung	158
3.6.3	Ausgabevalidierung	164
3.6.4	Schutz von Anwendungsparametern	171
3.6.5	Sonderfall Dateiuploads	172
3.6.6	Sonderfall URLs	174
3.6.7	Sonderfall Kommandos	175
3.6.8	Sonderfall XML	175
3.6.9	Sonderfall HTML-Markup	176
3.6.10	Überblick und Empfehlungen	178
3.7	Identifikation & Registrierung	179
3.7.1	Benutzerkennungen	179
3.7.2	Besucher-Tracking	180
3.7.3	IP-Adressen	181
3.7.4	Benutzerregistrierung durch technische Identifikation	182
3.7.5	Benutzerregistrierung durch persönliche Identifikation	183
3.7.6	Vorregistrierung	184
3.7.7	Gewinnspiele und Abstimmungen	184
3.7.8	Überblick und Empfehlungen	184
3.8	Authentifizierungsverfahren	186
3.8.1	IP-Adressen	187
3.8.2	HTTP Basic und HTTP Digest	188
3.8.3	Form-based Authentication	189
3.8.4	SSL-Client-Zertifikate (X.509-Zertifikate)	190
3.8.5	One Time Tokens (OTTs)	191
3.8.6	Kerberos (Windows-Authentifizierung)	192
3.8.7	SAML	193
3.8.8	OpenID	195
3.8.9	Mehrstufige Authentifizierung	196
3.8.10	Inter-Komponenten-Authentifizierung	198
3.8.11	Überblick und Empfehlungen	200
3.9	Benutzerpasswörter und Anmeldedialog	202
3.9.1	Passwortstärke	203
3.9.2	Generierte Passwörter	205
3.9.3	Passwort-Stärke-Funktionen	205
3.9.4	Neusetzen von Passwörtern (Passwort Reset)	207

3.9.5	Speicherung von Passwörtern (Passwort Hashing)	208
3.9.6	Überblick und Empfehlungen	210
3.10	Absicherung des Session Managements	212
3.10.1	Zufälligkeit der Session-ID	213
3.10.2	Härtung des Session Managements	213
3.10.3	Persistente Sessions	215
3.10.4	Session Binding (Browser Fingerprints)	216
3.10.5	Session-State-Kontrolle (CSRF- und Replay-Schutz)	216
3.10.6	Session Timeout	218
3.10.7	Mehrfachanmeldung (Concurrent Sessions)	220
3.10.8	Der Session-ID-Lifecycle	220
3.10.9	Shared Sessions	220
3.10.10	Überblick und Empfehlungen	221
3.11	Anti-Automatisierung	222
3.11.1	Limits	222
3.11.2	Verzögerungen	223
3.11.3	CAPTCHAS	223
3.11.4	Mehrstufigkeit	226
3.11.5	Weitere Verfahren	227
3.11.6	Überblick und Empfehlungen	228
3.12	Zugriffsschutz	230
3.12.1	Zugriffsmodelle	230
3.12.2	Mehrschichtige Zugriffskontrolle (expliziter Schutz)	231
3.12.3	Mehrschichtige Separierung (impliziter Schutz)	234
3.12.4	Rollen und Berechtigungen	235
3.12.5	Cross-Origin-Zugriffe	238
3.12.6	Access Tokens	243
3.12.7	OAuth	246
3.12.8	Geräte- bzw. Browser-Autorisierung	251
3.12.9	Überblick und Empfehlungen	252
3.13	Ereignisbehandlung	253
3.13.1	Fehlerbehandlung	254
3.13.2	Angriffserkennung und -behandlung (IDS/IPS)	255
3.13.3	User Alerting	257
3.13.4	Überblick und Empfehlungen	257
3.14	Datensicherheit und Datenschutz	258
3.14.1	Allgemeine Empfehlungen zum Schutz personenbezogener Daten	258
3.14.2	Datenverschlüsselung	260
3.14.3	Schutz der Integrität von Daten	264
3.14.4	Datenbehandlungsvorgaben	265
3.14.5	Datenschutzprofile	267

3.14.6	Abwicklung von Bezahlprozessen	268
3.14.7	Überblick und Empfehlungen	269
3.15	Clientseitige Sicherheitsaspekte	270
3.15.1	Gestaltung der Oberfläche	270
3.15.2	HTTP Strict Transport Security (HSTS)	272
3.15.3	Frame Busting	273
3.15.4	Schutz vor eingebettetem Schadcode (Werbebanner etc.)	275
3.15.5	Browserseitige XSS-Filter	276
3.15.6	Content Security Policy (CSP)	277
3.15.7	Dateidownloads	280
3.15.8	Browser-Plugins: Adobe Flash, Silverlight, ActiveX und Java-Applets	281
3.15.9	Visualisierung von Browsersicherheit	282
3.15.10	Überblick und Empfehlungen	282
3.16	Sicherheit von webbasierten Diensten	284
3.16.1	SOAP	285
3.16.2	XML-RPC	287
3.16.3	REST	287
3.16.4	JSON (Ajax)	288
3.16.5	WebSockets	289
3.16.6	RTMP	291
3.16.7	Überblick und Empfehlungen	292
3.17	Absicherung der Plattform	293
3.17.1	Generelle Architekturempfehlung der Infrastruktur	293
3.17.2	Härtung des SSL/TLS-Stacks	294
3.17.3	Härtung des Webserver	296
3.17.4	Laufzeitumgebung und Codeprivilegien	296
3.17.5	Webplattformen (WCMS-Systeme, Foren etc.)	302
3.17.6	Separierung (Abschottung)	304
3.17.7	Webanwendungsfirewalls (WAFs)	305
3.17.8	Cloud Computing	309
3.17.9	Überblick und Empfehlungen	311
3.18	Zusammenfassung	312
4	Sicherheitsuntersuchungen von Webanwendungen	315
4.1	Begriffe und Konzepte	315
4.1.1	Sicherheitsreview vs. Sicherheitstest	315
4.1.2	Risiko-basiertes Testen	316
4.1.3	Software Assurance (SwA)	316
4.1.4	Assurancegrad	316
4.1.5	Timeboxing	318
4.1.6	Low Hanging Fruits	319

4.2	Relevante Standards und Projekte	319
4.2.1	OSSTMM	319
4.2.2	OWASP ASVS Standard	319
4.2.3	OWASP Testing Guide	320
4.3	Lifecycle Security Testing	321
4.4	Wirksamkeit von Tools	323
4.5	Bewertungsverfahren	324
4.5.1	Risiken	324
4.5.2	CVSS	331
4.5.3	CWSS	332
4.5.4	DREAD	333
4.5.5	Bug Bars	335
4.5.6	Eignung der einzelnen Verfahren	336
4.6	Dynamische Testverfahren (Anwendungstests)	337
4.6.1	Verifikation von Sicherheitsvorgaben	338
4.6.2	Pentest (Penetrationstest)	340
4.6.3	Web Security Scanner (DAST)	346
4.6.4	Fault Injection.(Fuzzing)	350
4.6.5	Security Integration Tests	352
4.6.6	Deployment Reviews (Konfigurationsanalysen)	353
4.7	Statische Testverfahren (Security Codeanalysen)	354
4.7.1	Security Codescanner (SAST)	354
4.7.2	Code Firewalls	361
4.7.3	Security Unit Tests	361
4.7.4	Security Code Review	363
4.8	Validierung von Sicherheitsanforderungen	365
4.9	Architekturelle Sicherheitsanalysen	367
4.9.1	Erstellen einer Übersicht der Sicherheitsarchitektur	368
4.9.2	Analyse der architekturellen Vertrauensbeziehungen	369
4.9.3	Weitere mögliche Analyseinhalte	370
4.10	Bedrohungs- und Risikoanalysen (Risk & Threat Assessments)	374
4.10.1	Ermittlung von Bedrohungen, Gefährdungen und Risiken	374
4.10.2	Existierende Vorgehensweisen	376
4.10.3	Ein generisches Vorgehensmodell	377
4.10.4	Verfahren zur Bedrohungsidentifikation	380
4.10.5	Mapping von Gegenmaßnahmen	383
4.10.6	Bedrohungskataloge (Threat Intelligence)	385
4.10.7	Übergang zur Risikoanalyse	387
4.10.8	Tools	388
4.10.9	Weitere Aspekte	388
4.11	Zusammenfassung	389

5 Organisatorische (Web-)Anwendungssicherheit	393
5.1 Begriffe und Konzepte	395
5.1.1 Application Security Management	395
5.1.2 Secure Software Development Lifecycle (SSDLC)	396
5.1.3 Reifegrade und Reifegradmodelle	397
5.2 Relevante Standards und Projekte	399
5.2.1 BSI Leitfäden zur Entwicklung sicherer Webanwendungen	399
5.2.2 OpenSamm	399
5.2.3 BSIMM	400
5.2.4 ISO/IEC 27034	402
5.2.5 TSS-WEB	402
5.3 Maßnahmen zum Wissensaufbau und Wissenstransfer	402
5.3.1 Awareness und Management Attention schaffen	403
5.3.2 Schulungen durchführen	403
5.3.3 Multiplikatoren coachen	405
5.3.4 Lessons Learned und projektbezogene Workshops durchführen	405
5.3.5 Security Knowledge Base aufbauen	405
5.3.6 Mitarbeiter zertifizieren	406
5.4 Maßnahmen für das IT-Sicherheitsmanagement	406
5.4.1 Application Security Management Systems (ASMS) aufbauen	407
5.4.2 Richtlinien zur Anwendungssicherheit erstellen und pflegen	408
5.4.3 Vorgaben und Sign-Offs für Softwarelieferanten etablieren	409
5.4.4 Security Gates in Entwicklungs- und Beschaffungsprozesse integrieren	412
5.4.5 Risikomanagement für Anwendungssicherheit etablieren	414
5.4.6 Geschäftskritikalität und Schutzbedarf für Anwendungen spezifizieren	414
5.4.7 Zuständigkeiten für Anwendungssicherheit etablieren	415
5.4.8 Application Portfolio Management (APM) aufbauen	417
5.4.9 Project Portfolio Management (PPM) erweitern	418
5.4.10 Periodische Sicherheitsprüfungen aller Anwendungen durchführen	418
5.4.11 Application Incident und Problem Management einführen („Security Response“)	419
5.4.12 Key-Performance-Indikatoren (KPIs) definieren und reporten	420
5.5 Maßnahmen für das Projektmanagement	421
5.5.1 Zuständigkeiten und Sichtbarkeit für Sicherheit schaffen	422
5.5.2 Vertrauenswürdigkeit von Entwicklern sicherstellen	422
5.5.3 Fachliche Sicherheitsanforderungen und Risiken ermitteln	423
5.5.4 Sicherheitsumsetzungspläne erstellen und abstimmen	424
5.5.5 Sicherheitskonzepte erstellen und fortschreiben	424
5.5.6 Sicherheitskomponenten entkoppeln	425

5.5.7	Sicherheit im Change Management adressieren	426
5.5.8	Sicherheit in agile Vorgehensmodelle integrieren	427
5.5.9	Lessons Learned am Ende von Projekten durchführen	430
5.6	Maßnahmen für die Architektur	430
5.6.1	Architekturelle Vorgaben etablieren und kontrollieren	430
5.6.2	Sicherheitsdienste zentral bereitstellen	431
5.6.3	Technologiestacks und Blueprints vorgeben („Secure Foundation“)	432
5.6.4	Vorgaben und Sign-Offs für neue Technologien etablieren	433
5.7	Maßnahmen für die Entwicklung	436
5.7.1	Anreize schaffen	436
5.7.2	Software Security Group (SSG) aufbauen	436
5.7.3	Secure Coding Guidelines erstellen und pflegen	437
5.7.4	Vertrauen in die Entwicklungsumgebungen gewährleisten („Trusted Ecosystem“)	438
5.7.5	Sicherheitsfunktionen über Security-APIs zentral bereitstellen ...	439
5.7.6	Sicherheit bei Verwaltung von Sourcecode und Abhängigkeiten gewährleisten	439
5.7.7	Security Codeanalysen mit jedem Build durchführen	441
5.7.8	Sicherheit im Deployment-Prozess berücksichtigen	442
5.7.9	Kollaborative Sicherheitsreviews durchführen	442
5.7.10	Hacker-Techniken und -Tools einsetzen	442
5.7.11	Sicherheitsaspekte dokumentieren	443
5.8	Maßnahmen für die Qualitätssicherung	443
5.8.1	Produktiv- von Nicht-Produktiv-Systemen abschotten	444
5.8.2	Sicherheitskennzahlen spezifizieren und überwachen	444
5.8.3	Security Testing Factory aufbauen	446
5.8.4	Infrastruktur für Security-Scans bereitstellen	448
5.8.5	Sicherheitstests in verwendete Toolsuiten integrieren	449
5.8.6	Testbarkeit gewährleisten	450
5.8.7	Sicherheitstestplanung einfordern	450
5.9	Maßnahmen für den Betrieb	451
5.9.1	Plattform absichern	451
5.9.2	Laufende Sicherheitsscans der Plattform durchführen	451
5.9.3	Patch Management einbeziehen für 3rd-Party-Komponenten etablieren	452
5.9.4	Virtuelles Patch Management und Application Incident Response gewährleisten	453
5.9.5	Application IDS betreiben (Security Logging & Monitoring)	454
5.9.6	Zertifikatsmanagement betreiben	456
5.10	Maßnahmen für Softwarehersteller	456

5.11 Durchführen eines Programms zur Steigerung der Anwendungssicherheit (ASP)	457
5.11.1 Erfolgsfaktoren	457
5.11.2 Formelles Vorgehensmodell	458
5.11.3 Empfehlung für die praktische Durchführung eines ASPs	462
5.12 Zusammenfassung	464
6 Schlussbemerkungen	469
Erratum	E1
Anhang: Mapping von Maßnahmen zur OWASP Top 10	471
Glossar	473
Literatur	491
Sachverzeichnis	499