

Inhalt

Geleitwort des Fachgutachters	15
-------------------------------------	----

1 Einleitung 17

1.1 An wen richtet sich dieses Buch?	17
1.2 Was ist der Zweck dieses Buches?	20
1.3 Was leistet dieses Buch nicht	22
1.4 Wie ist dieses Buch aufgebaut?	25
1.4.1 Einleitung und Grundlagen	25
1.4.2 Sichere Webapplikationen bauen	26
1.4.3 Angriff und Verteidigung	26
1.4.4 Nützliches und Interessantes	27
1.5 Über die Autoren	27
1.5.1 Mario Heiderich	27
1.5.2 Christian Matthies	27
1.5.3 fukami	27
1.5.4 Johannes Dahse	27

2 Rechtslage 29

2.1 § 202c – Der Hackerparagraph	29
2.2 Erste Konsequenzen und Folgen	31
2.2.1 Wir sind dann mal weg	31
2.2.2 Das BSI im juristischen Fadenkreuz	33
2.2.3 Kriminell um jeden Preis	35
2.2.4 Das EICAR-Positionspapier	36
2.2.5 Fazit?	36
2.3 Wie verhält man sich nun am besten?	37
2.3.1 Darf man so was überhaupt?	38
2.3.2 Kommt darauf an	38
2.3.3 Manchmal ist es sogar erwünscht	39
2.3.4 Fazit	41
2.4 Ein Blick in die rechtliche Zukunft	42
2.4.1 Zusammenfassung	43

3	Vergangene Angriffe und Hacks	45
3.1	Samy – Der Wurm, der keiner sein wollte	45
3.1.1	Wie alles begann	46
3.1.2	Technische Grundlagen des Angriffs	47
3.1.3	Wie die Geschichte endete	49
3.2	Yamanner – Mailworming mit XSS	50
3.2.1	Die Vorgeschichte	50
3.2.2	Wie Yamanner funktionierte	51
3.2.3	Konsequenzen	52
3.3	Nduja Connection – XWW made in Italy	53
3.3.1	XWWie bitte?	54
3.3.2	Der eigentliche Wurm	54
3.3.3	Wie ging es weiter?	55
3.4	Gajaworm – Online-Games als Zielscheibe	55
3.4.1	Ein halb-reflektiver Wurm	56
3.4.2	Ist reflektives XSS ungefährlich?	57
3.5	Phishing – Das älteste Handwerk der Welt	58
3.5.1	Wie alles begann	58
3.5.2	A Phisher's bag of tricks	59
3.5.3	Homographische Angriffe und IDNs	62
3.5.4	Phishing und XSS	64
3.5.5	Redirects – Sich selbst phishende Seiten?	67
3.5.6	Fazit	68
3.6	Deine Post ist meine Post	70
3.7	Fazit	72
3.7.1	Zusammenfassung	73
4	Sicherheit im Web 2.0	75
4.1	Das neue Web	75
4.2	Privatsphäre im Social Web	76
4.2.1	Ein verändertes Nutzerverhalten	76
4.2.2	Wie sicher ist das (Social) Web 2.0 wirklich?	77
4.2.3	Auswirkungen auf Nutzer und Unternehmen	83
4.3	Gefahr aus der Wolke	85
4.3.1	Dabble DB – Datenbanken für alle	85
4.3.2	PHP per URL – freehostia.com	88
4.3.3	OnlyWire – Bookmarking mal anders	89
4.3.4	Sicherheitslücken mit der Google Code Search Engine googeln	93

4.3.5	OpenKapow – Angriff der Roboterkrieger	96
4.3.6	Das Internet als Payload	98
4.4	Ajax Security	101
4.4.1	XHR	103
4.4.2	Die Same Origin Policy	106
4.4.3	Das X in Ajax	109
4.4.4	JSON statt XML	111
4.4.5	Das Problem mit den Headern	112
4.4.6	Die Perle in der JSON-Auster	114
4.4.7	Probleme mit Ajax-Libraries	116
4.4.8	Fazit	117
4.4.9	Zusammenfassung	120

5 Webentwicklung mit Adobe Flash 121

5.1	Die Geschichte von Flash	122
5.2	Acronym Soup	124
5.3	Die Fähigkeiten von Flash	125
5.4	Aufruf und Einbindung	127
5.4.1	Parameter und Attribute	127
5.5	Die Sicherheitsmechanismen in Flash	130
5.5.1	Verantwortliche für die Sicherheit von Flash	131
5.5.2	Administrative Sicherheitseinstellungen	133
5.5.3	Sicherheitseinstellungen durch den User	137
5.5.4	Sicherheitseinstellungen durch Entwickler	143
5.5.5	Sandbox Security Model	145
5.5.6	Mögliche Sandboxes	147
5.5.7	Netzwerk-Protokolle	148
5.5.8	Port Blocking	148
5.5.9	Cross Domain Policies	149
5.6	ActionScript	156
5.6.1	Die Unterschiede zwischen AS2 und AS3	157
5.6.2	Kritische ActionScript-Funktionen	158
5.7	Daten aus Flash auf dem Server speichern	174
5.8	Werkzeuge zum Testen von Flash-Anwendungen	177
5.9	Angriffe auf Clients mithilfe des Flash-Plug-ins	184
5.10	Sinn und Unsinn von Obfuscation	186
5.11	Ausblick auf zukünftige Flash-Versionen	187
5.12	Zusammenfassung	188
5.13	Links	189

6 Sichere Webapplikationen bauen 191

6.1	Einleitung	191
6.2	Wichtige Grundlagen	192
6.2.1	Das HTTP-Protokoll	192
6.2.2	Encoding	206
6.2.3	Entities verstehen und nutzen	221
6.2.4	Was versteht man unter Filtering?	229
6.2.5	Warum Stripping selten sinnvoll ist	233
6.2.6	Reguläre Ausdrücke	237
6.2.7	Zusammenfassung	250
6.3	Planungs- und Designphase	250
6.3.1	Datenbankstruktur	253
6.3.2	Die Datenbank weiß, wer was darf	259
6.3.3	ACL im Detail	261
6.3.4	Backend und Pflegeskripte härten	266
6.3.5	Keine unnötige Preisgabe von Informationen	269
6.3.6	Zusammenfassung	274
6.4	Die Implementationsphase	274
6.4.1	GET-Parameter und Formulare	277
6.4.2	Validierung – A und O der Absicherung	278
6.4.3	Escapen	288
6.4.4	Filtering und Encoding	291
6.4.5	Links und Formulare gegen CSRF schützen	293
6.4.6	Zufallszahlen – aber richtig	298
6.4.7	CAPTCHAs – Sinn und Unsinn der Menscherkennung	299
6.4.8	Zusammenfassung	304
6.5	Sichere Datei-Uploads	304
6.5.1	Verbreitete Sicherheitslücken	305
6.5.2	Schutzmaßnahmen	315
6.5.3	Zusammenfassung	321
6.6	Kontaktformulare und Form-Mailer	321
6.6.1	Aufbau einer Mail	322
6.6.2	Header Injections	323
6.6.3	Weitere Risiken	327
6.6.4	Zusammenfassung	329
6.7	Redirects	329
6.7.1	Redirects per HTML	331
6.7.2	Redirects per JavaScript	332
6.7.3	Die Weiterleitung ins Grauen	333
6.7.4	HRS und die Kröte auf dem Grund des Brunnens	337

6.7.5	Immer und immer wieder	339
6.7.6	Redirects sicher implementieren	340
6.7.7	Zusammenfassung	344
6.8	Includes, Pfade und Konfigurationen	344
6.8.1	Local File Inclusions	347
6.8.2	Includes von fremden Servern	348
6.8.3	Vorsicht vor weiteren Include-Methoden	351
6.8.4	Schutzmaßnahmen	351
6.8.5	Ordner-Relikte und Backups	357
6.8.6	Zusammenfassung	360
6.9	Eval, Shell-Methoden und User Generated Code	360
6.9.1	Serverseitiges eval()	362
6.9.2	Clientseitiges eval()	364
6.9.3	Schutzmaßnahmen	364
6.9.4	User Generated Code – Geht das überhaupt?	366
6.9.5	Zusammenfassung	371
6.10	Sessions	371
6.10.1	Was genau sind eigentlich Sessions?	372
6.10.2	Offensichtliche Fehlerquellen	373
6.10.3	Session Fixation	376
6.10.4	Mehr Sicherheitsrelevantes zu Sessions	381
6.10.5	Zusammenfassung	384
6.11	Cookies	384
6.11.1	Sind Cookies Würmer?	385
6.11.2	Der Aufbau eines Cookies	387
6.11.3	Cookies und Domains	389
6.11.4	Cookies und JavaScript	390
6.11.5	HTTPOOnly als Rettung?	392
6.11.6	Fast tadellos	393
6.11.7	Was bleibt zur Defensive?	393
6.11.8	Zusammenfassung	394
6.12	Login und Authentifizierung	395
6.12.1	Information Disclosure	396
6.12.2	XSS im Login-Formular	399
6.12.3	SQL Injections in Login-Formularen	401
6.12.4	Mir nach, User!	405
6.12.5	Apropos Cookies und Logins	407
6.12.6	Schutzmaßnahmen	408
6.12.7	Zusammenfassung	411

6.13	WYSIWYG-Editoren	411
6.13.1	Wie WYSIWYG-Editoren funktionieren	414
6.13.2	WYSIWYG und XSS	415
6.13.3	WYSIWYG – aber bitte sicher	418
6.13.4	WYSIWYG Editor of Death	420
6.13.5	Zusammenfassung	422
6.14	Feeds	422
6.15	Verbreitete Sicherheitslücken	424
6.16	Lokale Exploits und Chrome	426
6.16.1	Zusammenfassung	430
6.17	Fehlermeldungen	430
6.17.1	Zusammenfassung	436
7 Testphase		437
7.1	Die eigene Applikation hacken	437
7.2	Manuelles Vorgehen	437
7.2.1	Source Code Reviews	441
7.3	Automatisiertes Vorgehen	444
8 Pflege- und Erweiterungsphase		451
8.1	Monitoring und Logging	452
8.2	Bestehende Applikationen absichern	456
8.2.1	Eine Datei, sie alle zu filtern	457
8.3	Plug-ins, Extensions und Themes	461
8.3.1	Zusammenfassung	463
9 XSS		465
9.1	Was ist XSS?	465
9.2	Kontextsensitives Schadpotential	467
9.3	XSS-Würmer	469
9.4	XSS in allen Facetten	475
9.4.1	Reflektives XSS	476
9.4.2	Persistentes XSS	484
9.4.3	Lazy-XSS – Angriffe auf das Backend	488
9.4.4	Untraceable XSS – Der unsichtbare Exploit	492
9.4.5	XSS per Stylesheet	498
9.4.6	XSS via XXE und UTF-7 ohne UTF-7	502
9.4.7	Zusammenfassung	504

10 Cross Site Request Forgeries 505

10.1	CSRF und XSS	508
10.1.1	Anti-XSRF-Schutzmaßnahmen vs. XSS	509
10.1.2	Exploiting Anti-XSRF geschütztes XSS	509
10.1.3	Exploiting Logged-Out XSS	510
10.2	Lesende Requests und Information Disclosure	515
10.2.1	Zustandschecks mit JavaScript	515
10.2.2	JavaScript Hijacking	516
10.2.3	Schutzmaßnahmen	519
10.3	Real Life Examples	521
10.3.1	Der Amazon-Exploit von Chris Shiflett	521
10.3.2	Der Gmail-Exploit von pdp	522
10.3.3	Der Gmail-Exploit von Jeremiah Grossman	522

11 SQL Injection 525

11.1	Vorgehensweise und Aufbau	526
11.2	Folgen eines Angriffs	528
11.2.1	Authentication Bypass	529
11.2.2	Informationsdiebstahl	530
11.2.3	Denial of Service	532
11.2.4	Datenmanipulation	533
11.2.5	Übernahme des Servers	535
11.3	Unterarten von SQL Injections	536
11.3.1	Blind SQL Injections	536
11.3.2	Stored Procedure Injection	544
11.4	Datenbanksystemspezifische SQL Injections	547
11.4.1	Fingerprinting des Datenbanksystems	547
11.4.2	Mapping der Datenbank	548
11.4.3	Angriffe auf das System	562
11.5	Umgehen von Filtern	572
11.5.1	Zusammenfassung	575

12 Directory Traversal 577

12.1	Schutzmaßnahmen mit zweifelhafter Wirkung	579
12.2	Code Execution via Directory Traversal	582
12.2.1	Zusammenfassung	583

13 RCE und LFI 585

13.1 Zusammenfassung 592

14 URI-Attacken 593

14.1 Der Browser als Gateway 595
14.2 Schutzmaßnahmen und Abwehr 599
 14.2.1 Zusammenfassung 601

15 Projekte und Tools 603

15.1 NoScript 603
15.2 HTML Purifier 606
15.3 ratproxy 609
15.4 PHPIDS 612
 15.4.1 Warum man das PHPIDS einsetzen sollte 614
 15.4.2 Anforderungen 615
 15.4.3 Installation und Benutzung 615
 15.4.4 Arbeiten mit dem Impact 618
 15.4.5 Logging und Ergebnisanalyse 619
 15.4.6 Allgemeine Angriffserkennung 621
 15.4.7 Performance 624
 15.4.8 Ausblick 625

Index 627