

# Web Application Defender's Cookbook

Battling  
Hackers and  
Protecting  
Users



■ Ryan C. Barnett



---

# **The Web Application Defender's Cookbook**



# The Web Application Defender's Cookbook

Battling Hackers and  
Protecting Users

Ryan Barnett



WILEY

Wiley Publishing, Inc.

---

**The Web Application Defender's Cookbook: Battling Hackers and Protecting Users**

Published by  
John Wiley & Sons, Inc.  
10475 Crosspoint Boulevard  
Indianapolis, IN 46256  
www.wiley.com

Copyright © 2013 by Ryan Barnett

Published simultaneously in Canada

ISBN: 978-1-118-36218-1  
ISBN: 978-1-118-56871-2 (ebk)  
ISBN: 978-1-118-41705-8 (ebk)  
ISBN: 978-1-118-56865-1 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Limit of Liability/Disclaimer of Warranty:** The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or website may provide or recommendations it may make. Further, readers should be aware that Internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Not all content that is available in standard print versions of this book may appear or be packaged in all book formats. If you have purchased a version of this book that did not include media that is referenced by or accompanies a standard print version, you may request this media by visiting <http://booksupport.wiley.com>. For more information about Wiley products, visit us at [www.wiley.com](http://www.wiley.com).

**Library of Congress Control Number:** 2012949513

**Trademarks:** Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

---

*This book is dedicated to my incredible daughter, Isabella. You are so full of imagination, kindness, and humor that I have a constant smile on my face. You are my Supergirl-flying, tae-kwon-do-kicking, fairy princess! I thank God every day for bringing you into my life and for allowing me the joy and privilege of being your father.*

*I love you Izzy, and I am so proud of you.*





---

# Credits

**Executive Editor**

Carol Long

**Project Editor**

Ed Connor

**Technical Editor**

Michael Gregg

**Production Editor**

Daniel Scribner

**Copy Editor**

Gayle Johnson

**Editorial Manager**

Mary Beth Wakefield

**Freelancer Editorial Manager**

Rosemarie Graham

**Associate Director of Marketing**

David Mayhew

**Marketing Manager**

Ashley Zurcher

**Business Manager**

Amy Knies

**Production Manager**

Tim Tate

**Vice President and Executive Group****Publisher**

Richard Swadley

**Vice President and Executive Publisher**

Neil Edde

**Associate Publisher**

Jim Minatel

**Project Coordinator, Cover**

Katie Crocker

**Composer**

Craig Johnson,  
Happenstance Type-O-Rama

**Proofreader**

Nicole Hirschman

**Indexer**

Ron Strauss

**Cover Image**

© Mak\_Art / iStockPhoto

**Cover Designer**

Ryan Sneed



---

# About the Author



**R**yan Barnett is renowned in the web application security industry for his unique expertise. After a decade of experience defending government and commercial web sites, he joined the Trustwave SpiderLabs Research Team. He specializes in application defense research and leads the open source ModSecurity web application firewall project.

In addition to his commercial work at Trustwave, Ryan is also an active contributor to many community-based security projects. He serves as the Open Web Application Security Project (OWASP) ModSecurity Core Rule Set project leader and is a contributor on the OWASP Top Ten and AppSensor projects. He is a Web Application Security Consortium Board Member and leads the Web Hacking Incident Database and the Distributed Web Honeypot projects. At the SANS Institute, he is a certified instructor and contributor on the Top 20 Vulnerabilities and CWE/SANS Top 25 Most Dangerous Programming Errors projects.

Ryan is regularly consulted by news outlets that seek his insights into and analysis of emerging web application attacks, trends, and defensive techniques. He is a frequent speaker and trainer at key industry events including Black Hat, SANS AppSec Summit, and OWASP AppSecUSA.



---

# About the Technical Editor



**M**ichael Gregg is the CEO of Superior Solutions, Inc. ([www.thesolutionfirm.com](http://www.thesolutionfirm.com)), a Houston-based IT security-consulting firm. His organization performs security assessments and penetration testing for Fortune 1000 firms. He is frequently cited by major and trade print publications as a cyber security expert. He has appeared as an expert commentator for network broadcast outlets and print publications and has spoken at major security conferences such as HackerHalted, Fusion, and CA World.



---

# Acknowledgments

I must begin by thanking my wonderful wife, Linda. When I came to her with the idea of writing another book, she was fully supportive even though she understood the sacrifice it would require. I thank her for her continued patience and for enduring many late nights of “angry typing.” She has always encouraged and supported me both professionally and personally. The completion of this book is not my accomplishment alone but our whole family’s because it was truly a team effort. I love you Linda and am honored that you are my partner for life.

I would like to thank Nick Percoco, Senior VP of Trustwave SpiderLabs, for his unwavering support of ModSecurity and for appointing me as its manager. I am fortunate to work with intelligent, clever and funny people. Unfortunately I cannot list them all here, however, I must single out Breno Silva Pinto. Breno is the lead developer of ModSecurity and we have worked closely on the project for 2 years. I am constantly impressed with his insights, ingenuity and technical skill for web application security. This book would not have been possible without Breno’s contributions to ModSecurity features and capabilities.

I would also like to thank two specific ModSecurity community members who epitomize the “giving back” philosophy of the open source community. Thanks to Christian Bockermann for developing many ModSecurity support tools such as the AuditConsole and to Josh Zlatin for always helping users on the mail-list and for contributions to the OWASP ModSecurity CRS.

Last but not least, I want to specifically thank OWASP members: Tom Brennan, Jim Manico, and Sarah Baso. Your tireless work ethic and commitment to the OWASP mission is undeniable. I would also like to thank Michael Coates for starting the AppSensor project and both Colin Watson and John Melton for expanding its capabilities.





---

# Contents

Foreword .....	.xix
Introduction .....	.xxiii
<b>I Preparing the Battle Space .....</b>	<b>1</b>
<b>1 Application Fortification .....</b>	<b>7</b>
<i>Recipe 1-1: Real-time Application Profiling .....</i>	<i>7</i>
<i>Recipe 1-2: Preventing Data Manipulation with         Cryptographic Hash Tokens .....</i>	<i>15</i>
<i>Recipe 1-3: Installing the OWASP ModSecurity Core Rule Set (CRS) .....</i>	<i>19</i>
<i>Recipe 1-4: Integrating Intrusion Detection System Signatures .....</i>	<i>33</i>
<i>Recipe 1-5: Using Bayesian Attack Payload Detection .....</i>	<i>38</i>
<i>Recipe 1-6: Enable Full HTTP Audit Logging .....</i>	<i>48</i>
<i>Recipe 1-7: Logging Only Relevant Transactions .....</i>	<i>52</i>
<i>Recipe 1-8: Ignoring Requests for Static Content .....</i>	<i>53</i>
<i>Recipe 1-9: Obscuring Sensitive Data in Logs .....</i>	<i>54</i>
<i>Recipe 1-10: Sending Alerts to a Central Log Host Using Syslog .....</i>	<i>58</i>
<i>Recipe 1-11: Using the ModSecurity AuditConsole .....</i>	<i>60</i>
<b>2 Vulnerability Identification and Remediation .....</b>	<b>67</b>
<i>Recipe 2-1: Passive Vulnerability Identification .....</i>	<i>70</i>
<i>Recipe 2-2: Active Vulnerability Identification .....</i>	<i>79</i>
<i>Recipe 2-3: Manual Scan Result Conversion .....</i>	<i>88</i>
<i>Recipe 2-4: Automated Scan Result Conversion .....</i>	<i>92</i>
<i>Recipe 2-5: Real-time Resource Assessments and Virtual Patching .....</i>	<i>99</i>
<b>3 Poisoned Pawns (Hacker Traps) .....</b>	<b>115</b>
<i>Recipe 3-1: Adding Honeypot Ports .....</i>	<i>116</i>
<i>Recipe 3-2: Adding Fake robots.txt Disallow Entries .....</i>	<i>118</i>
<i>Recipe 3-3: Adding Fake HTML Comments .....</i>	<i>123</i>
<i>Recipe 3-4: Adding Fake Hidden Form Fields .....</i>	<i>128</i>
<i>Recipe 3-5: Adding Fake Cookies .....</i>	<i>131</i>

**II Asymmetric Warfare . . . . . 137**

<b>4</b>	Reputation and Third-Party Correlation . . . . .	139
	<i>Recipe 4-1: Analyzing the Client's Geographic Location Data</i> . . . . .	141
	<i>Recipe 4-2: Identifying Suspicious Open Proxy Usage</i> . . . . .	147
	<i>Recipe 4-3: Utilizing Real-time Blacklist Lookups (RBL)</i> . . . . .	150
	<i>Recipe 4-4: Running Your Own RBL</i> . . . . .	157
	<i>Recipe 4-5: Detecting Malicious Links</i> . . . . .	160
<b>5</b>	Request Data Analysis . . . . .	171
	<i>Recipe 5-1: Request Body Access</i> . . . . .	172
	<i>Recipe 5-2: Identifying Malformed Request Bodies</i> . . . . .	178
	<i>Recipe 5-3: Normalizing Unicode</i> . . . . .	182
	<i>Recipe 5-4: Identifying Use of Multiple Encodings</i> . . . . .	186
	<i>Recipe 5-5: Identifying Encoding Anomalies</i> . . . . .	189
	<i>Recipe 5-6: Detecting Request Method Anomalies</i> . . . . .	193
	<i>Recipe 5-7: Detecting Invalid URI Data</i> . . . . .	197
	<i>Recipe 5-8: Detecting Request Header Anomalies</i> . . . . .	200
	<i>Recipe 5-9: Detecting Additional Parameters</i> . . . . .	209
	<i>Recipe 5-10: Detecting Missing Parameters</i> . . . . .	212
	<i>Recipe 5-11: Detecting Duplicate Parameter Names</i> . . . . .	214
	<i>Recipe 5-12: Detecting Parameter Payload Size Anomalies</i> . . . . .	216
	<i>Recipe 5-13: Detecting Parameter Character Class Anomalies</i> . . . . .	219
<b>6</b>	Response Data Analysis . . . . .	223
	<i>Recipe 6-1: Detecting Response Header Anomalies</i> . . . . .	224
	<i>Recipe 6-2: Detecting Response Header Information Leakages</i> . . . . .	234
	<i>Recipe 6-3: Response Body Access</i> . . . . .	238
	<i>Recipe 6-4: Detecting Page Title Changes</i> . . . . .	240
	<i>Recipe 6-5: Detecting Page Size Deviations</i> . . . . .	243
	<i>Recipe 6-6: Detecting Dynamic Content Changes</i> . . . . .	246
	<i>Recipe 6-7: Detecting Source Code Leakages</i> . . . . .	249
	<i>Recipe 6-8: Detecting Technical Data Leakages</i> . . . . .	253
	<i>Recipe 6-9: Detecting Abnormal Response Time Intervals</i> . . . . .	256
	<i>Recipe 6-10: Detecting Sensitive User Data Leakages</i> . . . . .	259
	<i>Recipe 6-11: Detecting Trojan, Backdoor, and Webshell Access Attempts</i> . . . . .	262

<b>7</b>	Defending Authentication . . . . .	265
	<i>Recipe 7-1: Detecting the Submission of Common/Default Usernames . . . .</i>	266
	<i>Recipe 7-2: Detecting the Submission of Multiple Usernames . . . . .</i>	269
	<i>Recipe 7-3: Detecting Failed Authentication Attempts . . . . .</i>	272
	<i>Recipe 7-4: Detecting a High Rate of Authentication Attempts . . . . .</i>	274
	<i>Recipe 7-5: Normalizing Authentication Failure Details . . . . .</i>	280
	<i>Recipe 7-6: Enforcing Password Complexity . . . . .</i>	283
	<i>Recipe 7-7: Correlating Usernames with SessionIDs . . . . .</i>	286
<b>8</b>	Defending Session State . . . . .	291
	<i>Recipe 8-1: Detecting Invalid Cookies . . . . .</i>	291
	<i>Recipe 8-2: Detecting Cookie Tampering . . . . .</i>	297
	<i>Recipe 8-3: Enforcing Session Timeouts . . . . .</i>	302
	<i>Recipe 8-4: Detecting Client Source Location Changes</i> <i>    During Session Lifetime . . . . .</i>	307
	<i>Recipe 8-5: Detecting Browser Fingerprint Changes During Sessions . . . .</i>	314
<b>9</b>	Preventing Application Attacks . . . . .	323
	<i>Recipe 9-1: Blocking Non-ASCII Characters . . . . .</i>	323
	<i>Recipe 9-2: Preventing Path-Traversal Attacks . . . . .</i>	327
	<i>Recipe 9-3: Preventing Forceful Browsing Attacks . . . . .</i>	330
	<i>Recipe 9-4: Preventing SQL Injection Attacks . . . . .</i>	332
	<i>Recipe 9-5: Preventing Remote File Inclusion (RFI) Attacks . . . . .</i>	336
	<i>Recipe 9-6: Preventing OS Commanding Attacks . . . . .</i>	340
	<i>Recipe 9-7: Preventing HTTP Request Smuggling Attacks . . . . .</i>	342
	<i>Recipe 9-8: Preventing HTTP Response Splitting Attacks . . . . .</i>	345
	<i>Recipe 9-9: Preventing XML Attacks . . . . .</i>	347
<b>10</b>	Preventing Client Attacks . . . . .	353
	<i>Recipe 10-1: Implementing Content Security Policy (CSP) . . . . .</i>	353
	<i>Recipe 10-2: Preventing Cross-Site Scripting (XSS) Attacks . . . . .</i>	362
	<i>Recipe 10-3: Preventing Cross-Site Request Forgery (CSRF) Attacks . . . .</i>	371
	<i>Recipe 10-4: Preventing UI Redressing (Clickjacking) Attacks . . . . .</i>	377
	<i>Recipe 10-5: Detecting Banking Trojan (Man-in-the-Browser) Attacks . . .</i>	381
<b>11</b>	Defending File Uploads . . . . .	387
	<i>Recipe 11-1: Detecting Large File Sizes . . . . .</i>	387
	<i>Recipe 11-2: Detecting a Large Number of Files . . . . .</i>	389
	<i>Recipe 11-3: Inspecting File Attachments for Malware . . . . .</i>	390

<b>12</b>	Enforcing Access Rate and Application Flows . . . . .	395
	<i>Recipe 12-1: Detecting High Application Access Rates.</i> . . . . .	395
	<i>Recipe 12-2: Detecting Request/Response Delay Attacks.</i> . . . . .	405
	<i>Recipe 12-3: Identifying Inter-Request Time Delay Anomalies</i> . . . . .	411
	<i>Recipe 12-4: Identifying Request Flow Anomalies</i> . . . . .	413
	<i>Recipe 12-5: Identifying a Significant Increase in Resource Usage</i> . . . . .	414
<b>III</b>	<b>Tactical Response.</b> . . . . .	<b>419</b>
<b>13</b>	Passive Response Actions. . . . .	421
	<i>Recipe 13-1: Tracking Anomaly Scores</i> . . . . .	421
	<i>Recipe 13-2: Trap and Trace Audit Logging</i> . . . . .	427
	<i>Recipe 13-3: Issuing E-mail Alerts</i> . . . . .	428
	<i>Recipe 13-4: Data Sharing with Request Header Tagging</i> . . . . .	436
<b>14</b>	Active Response Actions. . . . .	441
	<i>Recipe 14-1: Using Redirection to Error Pages</i> . . . . .	442
	<i>Recipe 14-2: Dropping Connections</i> . . . . .	445
	<i>Recipe 14-3: Blocking the Client Source Address</i> . . . . .	447
	<i>Recipe 14-4: Restricting Geolocation Access Through Defense Condition</i> <i>(DefCon) Level Changes.</i> . . . . .	452
	<i>Recipe 14-5: Forcing Transaction Delays</i> . . . . .	455
	<i>Recipe 14-6: Spoofing Successful Attacks</i> . . . . .	462
	<i>Recipe 14-7: Proxying Traffic to Honeypots</i> . . . . .	468
	<i>Recipe 14-8: Forcing an Application Logout.</i> . . . . .	471
	<i>Recipe 14-9: Temporarily Locking Account Access.</i> . . . . .	476
<b>15</b>	Intrusive Response Actions. . . . .	479
	<i>Recipe 15-1: JavaScript Cookie Testing</i> . . . . .	479
	<i>Recipe 15-2: Validating Users with CAPTCHA Testing</i> . . . . .	481
	<i>Recipe 15-3: Hooking Malicious Clients with BeEF</i> . . . . .	485
	<b>Index</b> . . . . .	<b>495</b>

---

# Foreword

A defender, the person responsible for protecting IT systems from being compromised, could just as easily be the first line of defense as the last line. In fact, a defender working for an average organization might be the *only* line of defense—the only thing standing between the bad guy and a headline-making data breach. Worse yet, perhaps the incident doesn't make headlines, and no one, including the defender, is the wiser.

Either way, when whatever crazy new Web 2.0 Ajax-laced HTML5-laden application has traversed the software development life cycle and successfully made it past the QA gate, when the third-party penetration testers are long gone, after management has signed off on all the security exceptions, and the application has been released to production, with or without the defender's knowledge or consent, "security" then becomes entirely the defender's responsibility. Rest assured that vulnerabilities will remain or will be introduced eventually. So, when all is said and done, a defender's mission is to secure the insecure, to identify incoming attacks and thwart them, and to detect and contain breaches.

That's why there should be no doubt about the importance of the role of a defender. Defenders often safeguard the personal data of millions of people. They may protect millions, perhaps billions, of dollars in online transactions and the core intellectual property of the entire business. You can bet that with so much on the line, with so much valuable information being stored, someone will want to steal it. And the bigger and more high profile the system, the more sustained and targeted the incoming attacks will be.

Making matters even more challenging, the bad guys have the luxury of picking their shots. They may attack a system whenever they want to, or not. A defender's job is 24/7/365, holidays, weekends, vacation days. The system must be ready, and the defender must be ready, at all times.

A defender's job description could read much like Ernest Shackleton's famous advertisement when he was looking for men to accompany him on his next Antarctic expedition:

*Men wanted for hazardous journey. Low wages, bitter cold, long hours of complete darkness. Safe return doubtful. Honour and recognition in event of success.*

A defender's success really comes down to understanding a few key points about the operational environment in which he or she works:

- Web sites are often deployed in such a way that they cannot be adequately mirrored in development, QA, or even staging. This means that the real and true security posture, the real and true risk to the business, can be fully grasped only when it hits production and becomes an actual risk. As such, defenders must be able to think on their feet, be nimble, and react quickly.
- Defenders will find themselves responsible for protecting web sites they did not create and have little or no insight into or control over. Management may not respect security and may be unwilling to fix identified vulnerabilities in a timely fashion, and that could be the long-term standard operating procedure. And maybe this is the right call, depending on business risk and the estimated cost of software security. Whatever the case may be, defenders must be able to identify incoming attacks, block as many exploits as they can, and contain breaches.
- Fighting fires and responding to daily threats must be an expected part of the role. Whether the business is fully committed to software security is immaterial, because software will always have vulnerabilities. Furthermore, everyone gets attacked eventually. A defender never wants to be late in seeing an attack and the last one to know about a breach. For a defender, attack identification and response time are crucial.
- Defenders, because they are on the front lines, learn a tremendous amount about the application's risk profile and the necessary security readiness required to thwart attackers. This intelligence is like gold when communicated to developers who are interested in creating ever more resilient systems. This intelligence is also like gold when informing the security assessment teams about what types of vulnerabilities they should focus on first when testing systems in either QA or production. Everyone needs actionable data. The best defenders have it.

Putting these practices to use requires specialized skills and experience. Normally, aspiring defenders don't get this type of how-to instruction from product README files or FAQs. Historically, the knowledge came from conversations with peers, blog posts, and mailing list conversations. Information scattered around the Internet is hard to cobble together into anything actionable. By the time you do, you might already have been hacked. Maybe that's why you picked up this book. Clearly web-based attackers are becoming more active and brazen every day, with no signs of slowing.

For a defender to be successful, there is simply no substitute for experience. And this kind of experience comes only from hour after hour, day after day, and year after year of being on the battlefield, learning what strategies and tactics work in a given situation.

This kind of experience certainly doesn't come quickly or easily. At the same time, this kind of information and the lessons learned can be documented, codified, and shared. This is what Ryan Barnett offers in this book: recipes for defense—recipes for success.

To all defenders, I leave you in Ryan's accomplished and capable hands. His reputation speaks for itself. Ryan is one of the original defenders. He has contributed more than anyone else in web security to define the role of the defender. And he's one of the best field practitioners I've ever seen. Good luck out there!

**Jeremiah Grossman**  
*Chief Technology Officer*  
*WhiteHat Security, Inc.*





- [Theory of Games and Economic Behavior \(60th Anniversary Edition\) pdf, azw \(kindle\)](#)
- [download Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians \(Chapman & Hall/CRC Texts in Statistical Science\) pdf, azw \(kindle\), epub](#)
- [read online Nagios 3 Enterprise Network Monitoring: Including Plug-Ins and Hardware Devices book](#)
- [read online Executive Functioning Workbook for Teens pdf](#)
  
- <http://www.gateaerospaceforum.com/?library/The-Complete-Short-Stories-of-Marcel-Proust.pdf>
- <http://studystategically.com/freebooks/Taken-by-Midnight--Midnight-Breed--Book-8-.pdf>
- <http://test1.batsinbelfries.com/ebooks/Nagios-3-Enterprise-Network-Monitoring--Including-Plug-Ins-and-Hardware-Devices.pdf>
- <http://toko-gumilar.com/books/Executive-Functioning-Workbook-for-Teens.pdf>