The Web Application Hacker's Handbook

Finding and Exploiting Security Flaws

2 Second Edition

Dafydd Stuttard Marcus Pinto

The Web Application Hacker's Handbook

Second Edition

Finding and Exploiting Security Flaws

Dafydd Stuttard
Marcus Pinto

WILEY
Wiley Publishing, Inc.

The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Second Edition

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

Published simultaneously in Canada

ISBN: 978-1-118-02647-2
ISBN: 978-1-118-17522-4 (ebk)
ISBN: 978-1-118-17524-8 (ebk)
ISBN: 978-1-118-17523-1 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Control Number: 2011934639

About the Authors

Dafydd Stuttard is an independent security consultant, author, and software developer. With more than 10 years of experience in security consulting, he specializes in the penetration testing of web applications and compiled software. Dafydd has worked with numerous banks, retailers, and other enterprises to help secure their web applications. He also has provided security consulting to several software manufacturers and governments to help secure their compiled software. Dafydd is an accomplished programmer in several languages. His interests include developing tools to facilitate all kinds of software security testing. Under the alias "PortSwigger," Dafydd created the popular Burp Suite of web application hacking tools; he continues to work actively on Burp's development. Dafydd is also cofounder of MDSec, a company providing training and consultancy on Internet security attack and defense. Dafydd has developed and presented training courses at various security conferences around the world, and he regularly delivers training to companies and governments. He holds master's and doctorate degrees in philosophy from the University of Oxford.

Marcus Pinto is cofounder of MDSec, developing and delivering training courses in web application security. He also performs ongoing security consultancy for financial, government, telecom, and retail verticals. His 11 years of experience in the industry have been dominated by the technical aspects of application security, from the dual perspectives of a consulting and end-user implementation role. Marcus has a background in attack-based security assessment and penetration testing. He has worked extensively with large-scale web application deployments in the financial services industry. Marcus has been developing and presenting database and web application training courses since 2005 at Black Hat and other worldwide security conferences, and for private-sector and government clients. He holds a master's degree in physics from the University of Cambridge.

About the Technical Editor

Dr. Josh Pauli received his Ph.D. in Software Engineering from North Dakota State University (NDSU) with an emphasis in secure requirements engineering and now serves as an Associate Professor of Information Security at Dakota State University (DSU). Dr. Pauli has published nearly 20 international journal and conference papers related to software security and his work includes invited presentations from the Department of Homeland Security and Black Hat Briefings. He teaches both undergraduate and graduate courses in system software security and web software security at DSU. Dr. Pauli also conducts web application penetration tests as a Senior Penetration Tester for an Information Security consulting firm where his duties include developing hands-on technical workshops in the area of web software security for IT professionals in the financial sector.

MDSec: The Authors' Company

Dafydd and Marcus are cofounders of MDSec, a company that provides training in attack and defense-based security, along with other consultancy services. If while reading this book you would like to put the concepts into practice, and gain hands-on experience in the areas covered, you are encouraged to visit our website, `http://mdsec.net`. This will give you access to hundreds of interactive vulnerability labs and other resources that are referenced throughout the book.

Credits

Executive Editor
Carol Long

Senior Project Editor
Adaobi Obi Tulton

Technical Editor
Josh Pauli

Production Editor
Kathleen Wisor

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

Proofreaders
Sarah Kaikini, Word One
Sheilah Ledwidge, Word One

Indexer
Robert Swanson

Cover Designer
Ryan Sneed

Cover Image
Wiley InHouse Design

Vertical Websites Project Manager
Laura Moss-Hollister

Vertical Websites Assistant Project Manager
Jenny Swisher

Vertical Websites Associate Producers
Josh Frank
Shawn Patrick
Doug Kuhn
Marilyn Hummel

Acknowledgments

We are indebted to the directors and others at Next Generation Security Software, who provided the right environment for us to realize the first edition of this book. Since then, our input has come from an increasingly wider community of researchers and professionals who have shared their ideas and contributed to the collective understanding of web application security issues that exists today. Because this is a practical handbook rather than a work of scholarship, we have deliberately avoided filling it with a thousand citations of influential articles, books, and blog postings that spawned the ideas involved. We hope that people whose work we discuss anonymously are content with the general credit given here.

We are grateful to the people at Wiley — in particular, to Carol Long for enthusiastically supporting our project from the outset, to Adaobi Obi Tulton for helping polish our manuscript and coaching us in the quirks of "American English," to Gayle Johnson for her very helpful and attentive copy editing, and to Katie Wisor's team for delivering a first-rate production.

A large measure of thanks is due to our respective partners, Becky and Amanda, for tolerating the significant distraction and time involved in producing a book of this size.

Both authors are indebted to the people who led us into our unusual line of work. Dafydd would like to thank Martin Law. Martin is a great guy who first taught me how to hack and encouraged me to spend my time developing techniques and tools for attacking applications. Marcus would like to thank his parents for everything they have done and continue to do, including getting me into computers. I've been getting into computers ever since.

Contents at a Glance

Contents

Introduction

This book is a practical guide to discovering and exploiting security flaws in web applications. By "web applications" we mean those that are accessed using a web browser to communicate with a web server. We examine a wide variety of different technologies, such as databases, file systems, and web services, but only in the context in which these are employed by web applications.

If you want to learn how to run port scans, attack firewalls, or break into servers in other ways, we suggest you look elsewhere. But if you want to know how to hack into a web application, steal sensitive data, and perform unauthorized actions, this is the book for you. There is enough that is interesting and fun to say on that subject without straying into any other territory.

Overview of This Book

The focus of this book is highly practical. Although we include sufficient background and theory for you to understand the vulnerabilities that web applications contain, our primary concern is the tasks and techniques that you need to master to break into them. Throughout the book, we spell out the specific steps you need to follow to detect each type of vulnerability, and how to exploit it to perform unauthorized actions. We also include a wealth of real-world examples, derived from the authors' many years of experience, illustrating how different kinds of security flaws manifest themselves in today's web applications.

Security awareness is usually a double-edged sword. Just as application developers can benefit from understanding the methods attackers use, hackers can gain from knowing how applications can effectively defend themselves. In addition to describing security vulnerabilities and attack techniques, we describe in detail the countermeasures that applications can take to thwart an

attacker. If you perform penetration tests of web applications, this will enable you to provide high-quality remediation advice to the owners of the applications you compromise.

Who Should Read This Book

This book's primary audience is anyone who has a personal or professional interest in attacking web applications. It is also aimed at anyone responsible for developing and administering web applications. Knowing how your enemies operate will help you defend against them.

We assume that you are familiar with core security concepts such as logins and access controls and that you have a basic grasp of core web technologies such as browsers, web servers, and HTTP. However, any gaps in your current knowledge of these areas will be easy to remedy, through either the explanations contained in this book or references elsewhere.

In the course of illustrating many categories of security flaws, we provide code extracts showing how applications can be vulnerable. These examples are simple enough that you can understand them without any prior knowledge of the language in question. But they are most useful if you have some basic experience with reading or writing code.

How This Book Is Organized

This book is organized roughly in line with the dependencies between the different topics covered. If you are new to web application hacking, you should read the book from start to finish, acquiring the knowledge and understanding you need to tackle later chapters. If you already have some experience in this area, you can jump straight into any chapter or subsection that particularly interests you. Where necessary, we have included cross-references to other chapters, which you can use to fill in any gaps in your understanding.

We begin with three context-setting chapters describing the current state of web application security and the trends that indicate how it is likely to evolve in the near future. We examine the core security problem affecting web applications and the defense mechanisms that applications implement to address this problem. We also provide a primer on the key technologies used in today's web applications.

The bulk of the book is concerned with our core topic — the techniques you can use to break into web applications. This material is organized around the key tasks you need to perform to carry out a comprehensive attack. These include mapping the application's functionality, scrutinizing and attacking its core defense mechanisms, and probing for specific categories of security flaws.

The book concludes with three chapters that pull together the various strands introduced in the book. We describe the process of finding vulnerabilities in an application's source code, review the tools that can help when you hack web applications, and present a detailed methodology for performing a comprehensive and deep attack against a specific target.

Chapter 1, "Web Application (In)security," describes the current state of security in web applications on the Internet today. Despite common assurances, the majority of applications are insecure and can be compromised in some way with a modest degree of skill. Vulnerabilities in web applications arise because of a single core problem: users can submit arbitrary input. This chapter examines the key factors that contribute to the weak security posture of today's applications. It also describes how defects in web applications can leave an organization's wider technical infrastructure highly vulnerable to attack.

Chapter 2, "Core Defense Mechanisms," describes the key security mechanisms that web applications employ to address the fundamental problem that all user input is untrusted. These mechanisms are the means by which an application manages user access, handles user input, and responds to attackers. These mechanisms also include the functions provided for administrators to manage and monitor the application itself. The application's core security mechanisms also represent its primary attack surface, so you need to understand how these mechanisms are intended to function before you can effectively attack them.

Chapter 3, "Web Application Technologies," is a short primer on the key technologies you are likely to encounter when attacking web applications. It covers all relevant aspects of the HTTP protocol, the technologies commonly used on the client and server sides, and various schemes used to encode data. If you are already familiar with the main web technologies, you can skim through this chapter.

Chapter 4, "Mapping the Application," describes the first exercise you need to perform when targeting a new application — gathering as much information as possible to map its attack surface and formulate your plan of attack. This process includes exploring and probing the application to catalog all its content and functionality, identifying all the entry points for user input, and discovering the technologies in use.

Chapter 5, "Bypassing Client-Side Controls," covers the first area of actual vulnerability, which arises when an application relies on controls implemented on the client side for its security. This approach normally is flawed, because any client-side controls can, of course, be circumvented. The two main ways in which applications make themselves vulnerable are by transmitting data via the client on the assumption that it will not be modified, and by relying on client-side checks on user input. This chapter describes a range of interesting technologies, including lightweight controls implemented within HTML, HTTP, and JavaScript, and more heavyweight controls using Java applets, ActiveX controls, Silverlight, and Flash objects.

Chapters 6, 7, and 8 cover some of the most important defense mechanisms implemented within web applications: those responsible for controlling user access. Chapter 6, "Attacking Authentication," examines the various functions by which applications gain assurance of their users' identity. This includes the main login function and also the more peripheral authentication-related functions such as user registration, password changing, and account recovery. Authentication mechanisms contain a wealth of different vulnerabilities, in both design and implementation, which an attacker can leverage to gain unauthorized access. These range from obvious defects, such as bad passwords and susceptibility to brute-force attacks, to more obscure problems within the authentication logic. We also examine in detail the types of multistage login mechanisms used in many security-critical applications and describe the new kinds of vulnerabilities these frequently contain.

Chapter 7, "Attacking Session Management," examines the mechanism by which most applications supplement the stateless HTTP protocol with the concept of a stateful session, enabling them to uniquely identify each user across several different requests. This mechanism is a key target when you are attacking a web application, because if you can break it, you can effectively bypass the login and masquerade as other users without knowing their credentials. We look at various common defects in the generation and transmission of session tokens and describe the steps you can take to discover and exploit these.

Chapter 8, "Attacking Access Controls," looks at the ways in which applications actually enforce access controls, relying on authentication and session management mechanisms to do so. We describe various ways in which access controls can be broken and how you can detect and exploit these weaknesses.

Chapters 9 and 10 cover a large category of related vulnerabilities, which arise when applications embed user input into interpreted code in an unsafe way. Chapter 9, "Attacking Data Stores," begins with a detailed examination of SQL injection vulnerabilities. It covers the full range of attacks, from the most obvious and trivial to advanced exploitation techniques involving out-of-band channels, inference, and time delays. For each kind of vulnerability and attack technique, we describe the relevant differences between three common types of databases: MS-SQL, Oracle, and MySQL. We then look at a range of similar attacks that arise against other data stores, including NoSQL, XPath, and LDAP.

Chapter 10, "Attacking Back-End Components," describes several other categories of injection vulnerabilities, including the injection of operating system commands, injection into web scripting languages, file path traversal attacks, file inclusion vulnerabilities, injection into XML, SOAP, back-end HTTP requests, and e-mail services.

Chapter 11, "Attacking Application Logic," examines a significant, and frequently overlooked, area of every application's attack surface: the internal logic it employs to implement its functionality. Defects in an application's logic are extremely varied and are harder to characterize than common vulnerabilities

such as SQL injection and cross-site scripting. For this reason, we present a series of real-world examples in which defective logic has left an application vulnerable. These illustrate the variety of faulty assumptions that application designers and developers make. From these different individual flaws, we derive a series of specific tests that you can perform to locate many types of logic flaws that often go undetected.

Chapters 12 and 13 cover a large and very topical area of related vulnerabilities that arise when defects within a web application can enable a malicious user of the application to attack other users and compromise them in various ways. Chapter 12, "Attacking Users: Cross-Site Scripting,", examines the most prominent vulnerability of this kind — a hugely prevalent flaw affecting the vast majority of web applications on the Internet. We examine in detail all the different flavors of XSS vulnerabilities and describe an effective methodology for detecting and exploiting even the most obscure manifestations of these.

Chapter 13, "Attacking Users: Other Techniques," looks at several other types of attacks against other users, including inducing user actions through request forgery and UI redress, capturing data cross-domain using various client-side technologies, various attacks against the same-origin policy, HTTP header injection, cookie injection and session fixation, open redirection, client-side SQL injection, local privacy attacks, and exploiting bugs in ActiveX controls. The chapter concludes with a discussion of a range of attacks against users that do not depend on vulnerabilities in any particular web application, but that can be delivered via any malicious web site or suitably positioned attacker.

Chapter 14, "Automating Customized Attacks," does not introduce any new categories of vulnerabilities. Instead, it describes a crucial technique you need to master to attack web applications effectively. Because every web application is different, most attacks are customized in some way, tailored to the application's specific behavior and the ways you have discovered to manipulate it to your advantage. They also frequently require issuing a large number of similar requests and monitoring the application's responses. Performing these requests manually is extremely laborious and prone to mistakes. To become a truly accomplished web application hacker, you need to automate as much of this work as possible to make your customized attacks easier, faster, and more effective. This chapter describes in detail a proven methodology for achieving this. We also examine various common barriers to the use of automation, including defensive session-handling mechanisms and CAPTCHA controls. Furthermore, we describe tools and techniques you can use to overcome these barriers.

Chapter 15, "Exploiting Information Disclosure," examines various ways in which applications leak information when under active attack. When you are performing all the other types of attacks described in this book, you should always monitor the application to identify further sources of information disclosure that you can exploit. We describe how you can investigate anomalous behavior and error messages to gain a deeper understanding of the application's

internal workings and fine-tune your attack. We also cover ways to manipulate defective error handling to systematically retrieve sensitive information from the application.

Chapter 16, "Attacking Native Compiled Applications," looks at a set of important vulnerabilities that arise in applications written in native code languages such as C and C++. These vulnerabilities include buffer overflows, integer vulnerabilities, and format string flaws. Because this is a potentially huge topic, we focus on ways to detect these vulnerabilities in web applications and look at some real-world examples of how these have arisen and been exploited.

Chapter 17, "Attacking Application Architecture," examines an important area of web application security that is frequently overlooked. Many applications employ a tiered architecture. Failing to segregate different tiers properly often leaves an application vulnerable, enabling an attacker who has found a defect in one component to quickly compromise the entire application. A different range of threats arises in shared hosting environments, where defects or malicious code in one application can sometimes be exploited to compromise the environment itself and other applications running within it. This chapter also looks at the range of threats that arise in the kinds of shared hosting environments that have become known as "cloud computing."

Chapter 18, "Attacking the Application Server," describes various ways in which you can target a web application by targeting the web server on which it is running. Vulnerabilities in web servers are broadly composed of defects in their configuration and security flaws within the web server software. This topic is on the boundary of the subjects covered in this book, because the web server is strictly a different component in the technology stack. However, most web applications are intimately bound up with the web server on which they run. Therefore, attacks against the web server are included in the book because they can often be used to compromise an application directly, rather than indirectly by first compromising the underlying host.

Chapter 19, "Finding Vulnerabilities in Source Code," describes a completely different approach to finding security flaws than those described elsewhere within this book. In many situations it may be possible to review an application's source code, not all of which requires cooperation from the application's owner. Reviewing an application's source code can often be highly effective in discovering vulnerabilities that would be difficult or time-consuming to detect by probing the running application. We describe a methodology, and provide a language-by-language cheat sheet, to enable you to perform an effective code review even if you have limited programming experience.

Chapter 20, "A Web Application Hacker's Toolkit," pulls together the various tools described in this book. These are the same tools the authors use when attacking real-world web applications. We examine the key features of these tools and describe in detail the type of work flow you generally need to employ to get the best out of them. We also examine the extent to which any fully automated tool

can be effective in finding web application vulnerabilities. Finally, we provide some tips and advice for getting the most out of your toolkit.

Chapter 21, "A Web Application Hacker's Methodology," is a comprehensive and structured collation of all the procedures and techniques described in this book. These are organized and ordered according to the logical dependencies between tasks when you are carrying out an actual attack. If you have read about and understood all the vulnerabilities and techniques described in this book, you can use this methodology as a complete checklist and work plan when carrying out an attack against a web application.

What's New in This Edition

In the four years since the first edition of this book was published, much has changed, and much has stayed the same. The march of new technology has, of course, continued apace, and this has given rise to specific new vulnerabilities and attacks. The ingenuity of hackers has also led to the development of new attack techniques and new ways of exploiting old bugs. But neither of these factors, technological or human, has created a revolution. The technologies used in today's applications have their roots in those that are many years old. And the fundamental concepts involved in today's cutting-edge exploitation techniques are older than many of the researchers who are applying them so effectively. Web application security is a dynamic and exciting area to work in, but the bulk of what constitutes our accumulated wisdom has evolved slowly over many years. It would have been distinctively recognizable to practitioners working a decade or more ago.

This second edition is not a complete rewrite of the first. Most of the material in the first edition remains valid and current today. Approximately 30% of the content in this edition is either new or extensively revised. The remaining 70% has had minor modifications or none at all. If you have upgraded from the first edition and feel disappointed by these numbers, you should take heart. If you have mastered all the techniques described in the first edition, you already have the majority of the skills and knowledge you need. You can focus on what is new in this edition and quickly learn about the areas of web application security that have changed in recent years.

One significant new feature of the second edition is the inclusion throughout the book of real examples of nearly all the vulnerabilities that are covered. Wherever you see a "Try It!" link, you can go online and work interactively with the example being discussed to confirm that you can find and exploit the vulnerability it contains. There are several hundred of these labs, which you can work through at your own pace as you read the book. The online labs are available on a subscription basis for a modest fee to cover the costs of hosting and maintaining the infrastructure involved.

If you want to focus on what's new in the second edition, here is a summary of the key areas where material has been added or rewritten:

Chapter 1, "Web Application (In)security," has been partly updated to reflect new uses of web applications, some broad trends in technologies, and the ways in which a typical organization's security perimeter has continued to change.

Chapter 2, "Core Defense Mechanisms," has had minor changes. A few examples have been added of generic techniques for bypassing input validation defenses.

Chapter 3, "Web Application Technologies," has been expanded with some new sections describing technologies that are either new or that were described more briefly elsewhere within the first edition. The topics added include REST, Ruby on Rails, SQL, XML, web services, CSS, VBScript, the document object model, Ajax, JSON, the same-origin policy, and HTML5.

Chapter 4, "Mapping the Application," has received various minor updates to reflect developments in techniques for mapping content and functionality.

Chapter 5, "Bypassing Client-Side Controls," has been updated more extensively. In particular, the section on browser extension technologies has been largely rewritten to include more detailed guidance on generic approaches to bytecode decompilation and debugging, how to handle serialized data in common formats, and how to deal with common obstacles to your work, including non-proxy-aware clients and problems with SSL. The chapter also now covers Silverlight technology.

Chapter 6, "Attacking Authentication," remains current and has only minor updates.

Chapter 7, "Attacking Session Management," has been updated to cover new tools for automatically testing the quality of randomness in tokens. It also contains new material on attacking encrypted tokens, including practical techniques for token tampering without knowing either the cryptographic algorithm or the encryption key being used.

Chapter 8, "Attacking Access Controls," now covers access control vulnerabilities arising from direct access to server-side methods, and from platform misconfiguration where rules based on HTTP methods are used to control access. It also describes some new tools and techniques you can use to partially automate the frequently onerous task of testing access controls.

The material in Chapters 9 and 10 has been reorganized to create more manageable chapters and a more logical arrangement of topics. Chapter 9, "Attacking Data Stores," focuses on SQL injection and similar attacks against other data store technologies. As SQL injection vulnerabilities have become more widely understood and addressed, this material now focuses more on practical situations where SQL injection is still found. There are also minor updates throughout to reflect current technologies and attack methods. A new section on using automated tools for exploiting SQL injection vulnerabilities is included. The material on LDAP injection has been largely rewritten to include more detailed

coverage of specific technologies (Microsoft Active Directory and OpenLDAP), as well as new techniques for exploiting common vulnerabilities. This chapter also now covers attacks against NoSQL.

Chapter 10, "Attacking Back-End Components," covers the other types of server-side injection vulnerabilities that were previously included in Chapter 9. New sections cover XML external entity injection and injection into back-end HTTP requests, including HTTP parameter injection/pollution and injection into URL rewriting schemes.

Chapter 11, "Attacking Application Logic," includes more real-world examples of common logic flaws in input validation functions. With the increased usage of encryption to protect application data at rest, we also include an example of how to identify and exploit encryption oracles to decrypt encrypted data.

The topic of attacks against other application users, previously covered in Chapter 12, has been split into two chapters, because this material was becoming unmanageably large. Chapter 12, "Attacking Users: Cross-Site Scripting," focuses solely on XSS. This material has been extensively updated in various areas. The sections on bypassing defensive filters to introduce script code have been completely rewritten to cover new techniques and technologies, including various little-known methods for executing script code on current browsers. There is also much more detailed coverage of methods for obfuscating script code to bypass common input filters. The chapter includes several new examples of real-world XSS attacks. A new section on delivering working XSS exploits in challenging conditions covers escalating an attack across application pages, exploiting XSS via cookies and the `Referer` header, and exploiting XSS in nonstandard request and response content such as XML. There is a detailed examination of browsers' built-in XSS filters and how these can be circumvented to deliver exploits. New sections discuss specific techniques for exploiting XSS in webmail applications and in uploaded files. Finally, there are various updates to the defensive measures that can be used to prevent XSS attacks.

The new Chapter 13, "Attacking Users: Other Techniques," unites the remainder of this huge area. The topic of cross-site request forgery has been updated to include CSRF attacks against the login function, common defects in anti-CSRF defenses, UI redress attacks, and common defects in framebusting defenses. A new section on cross-domain data capture includes techniques for stealing data by injecting text containing nonscripting HTML and CSS, and various techniques for cross-domain data capture using JavaScript and E4X. A new section examines the same-origin policy in more detail, including its implementation in different browser extension technologies, the changes brought by HTML5, and ways of crossing domains via proxy service applications. There are new sections on client-side cookie injection, SQL injection, and HTTP parameter pollution. The section on client-side privacy attacks has been expanded to include storage mechanisms provided by browser extension technologies and HTML5. Finally, a new section has been added drawing together general attacks against

web users that do not depend on vulnerabilities in any particular application. These attacks can be delivered by any malicious or compromised web site or by an attacker who is suitably positioned on the network.

Chapter 14, "Automating Customized Attacks," has been expanded to cover common barriers to automation and how to circumvent them. Many applications employ defensive session-handling mechanisms that terminate sessions, use ephemeral anti-CSRF tokens, or use multistage processes to update application state. Some new tools are described for handling these mechanisms, which let you continue using automated testing techniques. A new section examines CAPTCHA controls and some common vulnerabilities that can often be exploited to circumvent them.

Chapter 15, "Exploiting Information Disclosure," contains new sections about XSS in error messages and exploiting decryption oracles.

Chapter 16, "Attacking Native Compiled Applications," has not been updated.

Chapter 17, "Attacking Application Architecture," has a new section about vulnerabilities that arise in cloud-based architectures, and updated examples of exploiting architecture weaknesses.

Chapter 18, "Attacking the Application Server," contains several new examples of interesting vulnerabilities in application servers and platforms, including Jetty, the JMX management console, ASP.NET, Apple iDisk server, Ruby WEBrick web server, and Java web server. It also has a new section on practical approaches to circumventing web application firewalls.

Chapter 19, "Finding Vulnerabilities in Source Code," has not been updated.

Chapter 20, "A Web Application Hacker's Toolkit," has been updated with details on the latest features of proxy-based tool suites. It contains new sections on how to proxy the traffic of non-proxy-aware clients and how to eliminate SSL errors in browsers and other clients caused by the use of an intercepting proxy. This chapter contains a detailed description of the work flow that is typically employed when you test using a proxy-based tool suite. It also has a new discussion about current web vulnerability scanners and the optimal approaches to using these in different situations.

Chapter 21, "A Web Application Hacker's Methodology," has been updated to reflect the new methodology steps described throughout the book.

Tools You Will Need

This book is strongly geared toward hands-on techniques you can use to attack web applications. After reading the book, you will understand the specifics of each individual task, what it involves technically, and why it helps you detect and exploit vulnerabilities. The book is emphatically not about downloading a tool, pointing it at a target application, and believing what the tool's output tells you about the state of the application's security.

That said, you will find several tools useful, and sometimes indispensable, when performing the tasks and techniques we describe. All of these are available on the Internet. We recommend that you download and experiment with each tool as you read about it.

What's on the Website

The companion website for this book at `http://mdsec.net/wahh`, which you can also link to from `www/wiley.com/go/webhacker2e`, contains several resources that you will find useful in the course of mastering the techniques we describe and using them to attack actual applications. In particular, the website contains access to the following:

- Source code for some of the scripts we present in the book
- A list of current links to all the tools and other resources discussed in the book
- A handy checklist of the tasks involved in attacking a typical application
- Answers to the questions posed at the end of each chapter
- Hundreds of interactive vulnerability labs that are used in examples throughout this book and that are available on a subscription basis to help you develop and refine your skills

Bring It On

Web application security remains a fun and thriving subject. We enjoyed writing this book as much as we continue to enjoy hacking into web applications on a daily basis. We hope that you will also take pleasure from learning about the different techniques we describe and how you can defend against them.

Before going any further, we should mention an important caveat. In most countries, attacking computer systems without the owner's permission is against the law. The majority of the techniques we describe are illegal if carried out without consent.

The authors are professional penetration testers who routinely attack web applications on behalf of clients to help them improve their security. In recent years, numerous security professionals and others have acquired criminal records — and ended their careers — by experimenting on or actively attacking computer systems without permission. We urge you to use the information contained in this book only for lawful purposes.