Seven Deadliest
Web Application Attacks
Syngress Seven Deadliest Attacks Series

Seven Deadliest Microsoft Attacks
ISBN: 978-1-59749-551-6
Rob Kraus

Seven Deadliest Network Attacks
ISBN: 978-1-59749-549-3
Stacy Prowell

Seven Deadliest Social Network Attacks
ISBN: 978-1-59749-545-5
Carl Timm

Seven Deadliest Unified Communications Attacks
ISBN: 978-1-59749-547-9
Dan York

Seven Deadliest USB Attacks
ISBN: 978-1-59749-553-0
Brian Anderson

Seven Deadliest Web Application Attacks
ISBN: 978-1-59749-543-1
Mike Shema

Seven Deadliest Wireless Technologies Attacks
ISBN: 978-1-59749-541-7
Brad Haines

Visit www.syngress.com for more information on these titles and other resources.
Seven Deadliest Web Application Attacks

Mike Shema

Technical Editor Adam Ely
# Contents

About the Authors ..................................................................................................... ix
Introduction ............................................................................................................... xi

**CHAPTER 1 Cross-Site Scripting** ................................................................. 1
  Understanding HTML Injection ........................................................................ 2
  Identifying Points of Injection ......................................................................... 6
  Distinguishing Different Delivery Vectors ....................................................... 9
  Handling Character Sets Safely ...................................................................... 11
  Not Failing Secure ....................................................................................... 16
  Avoiding Blacklisted Characters Altogether ............................................... 16
  Dealing with Browser Quirks ........................................................................ 17
  The Unusual Suspects ................................................................................... 17
  Employing Countermeasures ........................................................................ 19
  Fixing a Static Character Set ........................................................................ 19
  Normalizing Character Sets and Encoding .................................................... 20
  Encoding the Output ..................................................................................... 21
  Beware of Exclusion Lists and Regexes ......................................................... 22
  Reuse, Don’t Reimplement, Code ................................................................. 23
  JavaScript Sandboxes .................................................................................... 24
  Summary ......................................................................................................... 25

**CHAPTER 2 Cross-Site Request Forgery** .................................................. 27
  Understanding Cross-Site Request Forgery .................................................... 28
  Request Forgery via Forced Browsing ............................................................ 30
  Attacking Authenticated Actions without Passwords .................................... 31
  Dangerous Liaison: CSRF and XSS ................................................................. 32
  Beyond GET ................................................................................................... 32
  Be Wary of the Tangled Web .......................................................................... 34
  Variation on a Theme: Clickjacking .............................................................. 34
  Employing Countermeasures ........................................................................ 36
  Defending the Web Application .................................................................... 37
  Defending the Web Browser .......................................................................... 43
  Summary ......................................................................................................... 44

**CHAPTER 3 Structured Query Language Injection** ..................................... 47
  Understanding SQL Injection ......................................................................... 49
  Breaking the Query ......................................................................................... 51
  Vivisection the Database ............................................................................... 57
  Alternate Attack Vectors ................................................................................ 60
Employing Countermeasures .............................................................. 60
Validating Input ................................................................................... 61
Securing the Query ............................................................................ 61
Protecting Information ........................................................................ 66
Stay Current with Database Patches ................................................... 67
Summary .............................................................................................. 68

**CHAPTER 4 Server Misconfiguration and Predictable Pages**

Understanding the Attacks ................................................................. 72
Identifying Insecure Design Patterns ................................................... 72
Targeting the Operating System ......................................................... 81
Attacking the Server ........................................................................... 86
Employing Countermeasures ............................................................. 87
Restricting File Access .................................................................... 87
Using Object References ................................................................. 88
Blacklisting Insecure Functions ......................................................... 88
Enforcing Authorization ................................................................. 89
Restricting Network Connections .................................................... 89
Summary .............................................................................................. 90

**CHAPTER 5 Breaking Authentication Schemes**

Understanding Authentication Attacks ............................................... 92
Replaying the Session Token ............................................................ 92
Brute Force ....................................................................................... 95
Sniffing ............................................................................................. 96
Resetting Passwords ........................................................................ 97
Cross-Site Scripting .......................................................................... 98
SQL Injection ..................................................................................... 98
Gulls and Gullibility .......................................................................... 99
Employing Countermeasures ........................................................... 100
Protect Session Cookies ..................................................................... 101
Engage the User ............................................................................... 102
Annoy the User ............................................................................... 102
Request Throttling ........................................................................... 103
Logging and Triangulation ............................................................ 104
Use Alternate Authentication Schemes ......................................... 105
Defeating Phishing ........................................................................... 106
Protecting Passwords ................................................................. 107
Summary .............................................................................................. 107
## CHAPTER 6  Logic Attacks ................................................................. 109
- Understanding Logic Attacks ................................................. 111
- Abusing Workflows .............................................................. 112
- Exploit Policies and Practices .............................................. 112
- Induction ............................................................................. 117
- Denial of Service .................................................................. 120
- Insecure Design Patterns .................................................... 121
- Information Sieves ............................................................... 124
- Employing Countermeasures ............................................... 125
  - Documenting Requirements .............................................. 125
  - Creating Robust Test Cases ............................................. 125
  - Mapping Policies to Controls ......................................... 127
  - Defensive Programming .................................................. 127
  - Verifying the Client ......................................................... 128
- Summary ............................................................................... 128

## CHAPTER 7  Web of Distrust ........................................................... 129
- Understanding Malware and Browser Attacks ...................... 130
  - Malware .......................................................................... 130
  - Plugging into Browser Plug-ins ...................................... 134
  - Domain Name System and Origins .................................. 135
  - HTML5 ............................................................................. 136
- Employing Countermeasures ............................................... 138
  - Safer Browsing ............................................................... 138
  - Isolating the Browser ....................................................... 139
  - DNS Security Extensions ............................................... 140
- Summary ............................................................................... 141

## Index .......................................................................................... 143

A preview chapter from *Seven Deadliest Microsoft Attacks* can be found after the index.
About the Authors

**Mike Shema** is the lead developer for the Web Application Scanning service offered by the vulnerability management company Qualys. The Web scanning service provides automated, accurate tests for most common Web vulnerabilities. Prior to Qualys, Mike gained extensive information security experience based on consulting work while at Foundstone. He has developed and conducted training on topics ranging from network security to wireless assessments to Web application penetration testing. Much of this experience has driven research into various security-related topics that he has presented at conferences in North America, Europe, and Asia, including BlackHat, InfoSec, and RSA.

Mike has also coauthored *Anti-Hacker Toolkit*, Third Edition and *Hacking Exposed: Web Applications*, Second Edition. He lives in San Francisco and would like to thank the RPG crew for keeping anachronistic random generators alive.

**Technical Editor**

**Adam Ely** (CISSP, NSA IAM, MCSE) is Director of Corporate Security for TiVo where he is responsible for IT security and corporate security policies. Adam has held positions with The Walt Disney Company where he was Manager of Information Security Operations for the Walt Disney Interactive Media Group, and Senior Manager of Technology for a Walt Disney acquired business. In addition, Adam was a consultant with Alvarez and Marsal where he led security engagements for clients. Adam’s background focuses on application and infrastructure security. Adam has published many application vulnerabilities, application security roadmaps, and other articles.
Introduction

INFORMATION IN THIS CHAPTER

- Book Overview and Key Learning Points
- Book Audience
- How This Book Is Organized
- Where to Go from Here

Pick your favorite cliche or metaphor you’ve heard regarding the Web. The aphorism might carry a generic description of Web security or generate a mental image of the threats and risks faced by and emanating from Web sites. This book attempts to cast a brighter light on the vagaries of Web security by tackling seven of the most, er, deadliest vulnerabilities that are exploited by attackers. Some of the attacks will sound very familiar. Other attacks may be unexpected, or seem uncommon simply because they aren’t on a top 10 list or don’t make headlines. Attackers often go for the lowest common denominator, which is why vulnerabilities such as cross-site scripting (XSS) and Structured Query Language (SQL) injection garner so much attention. Determined attackers also target the logic of a particular Web site – exploits that result in significant financial gain but have neither universal applicability from the attacker’s perspective nor universal detection mechanisms for the defender.

On the Web, information equals money. Credit cards clearly have value to attackers; underground e-commerce sites have popped up that deal in stolen cards. Yet our personal information, passwords, e-mail accounts, online game accounts, all have value to the right buyer. Then consider economic espionage and state-sponsored network attacks. It should be possible to map just about any scam, cheat, trick, ruse, and other synonyms from real-world conflict between people, companies, and countries to an attack that can be accomplished on the Web. There’s no lack of motivation for trying to gain illicit access to the wealth of information on the Web that isn’t intended to be public.

BOOK OVERVIEW AND KEY LEARNING POINTS

Each chapter in this book presents examples of different attacks conducted against Web sites. The methodology behind the attack is explored, as well as showing its potential impact. Then the chapter moves on to address possible countermeasures
Introduction

for different aspects of the attack. Countermeasures are a tricky beast. It’s important to understand how an attack works before a good defense can be designed. It’s also important to understand the limitations of a countermeasure and how other vulnerabilities might entirely bypass it. Security is an emergent property of the Web site; it’s not a summation of individual protections. Some countermeasures will show up several times, and others make only a brief appearance.

BOOK AUDIENCE

Anyone who uses the Web to check e-mail, shop, or work will benefit from knowing how the personal information on those sites might be compromised or even how familiar sites can harbor malicious content. Although most security relies on the site’s developers, consumers of Web applications can follow safe browsing practices to help protect their data.

Web application developers and security professionals will benefit from the technical details and methodology behind the Web attacks covered in this book. The first step to creating a more secure Web site is understanding the threats and risks of insecure code. Also, the chapters dive into countermeasures that can be applied to a site regardless of the programming language or technologies underpinning it.

Executive level management will benefit from understanding the threats to a Web site, and in many cases, how a simple attack — requiring nothing more than a Web browser — can severely impact a site. It should also illustrate that even though many attacks are simple to execute, good countermeasures require time and resources to implement properly. These points should provide strong arguments for allocating funding and resources to a site’s security to protect the wealth of information that Web sites manage.

This book assumes some basic familiarity with the Web. Web security attacks manipulate HTTP traffic to inject payloads or take advantage of deficiencies in the protocol. They also require understanding HTML to manipulate forms or inject code that puts the browser at the mercy of the attacker. This isn’t a prerequisite for understanding the broad strokes of an attack or learning how attackers compromise a site. For example, it’s good to know that HTTP uses port 80 by default for unencrypted traffic and port 443 for traffic encrypted with the Secure Sockets Layer (SSL). Sites use the https:// to designate SSL connections. Additional details are necessary for developers and security professionals who wish to venture deeper into the methodology of attacks and defense.

Readers already familiar with basic Web concepts can skip the next two sections.

One Origin to Rule Them All

Web browsers have gone through many iterations on many platforms: Konqeror, Mosaic, Mozilla, Internet Explorer, Opera, and Safari. Browsers have a rendering engine at their core. Microsoft calls IE’s engine Trident. Safari uses WebKit. Firefox
relies on Gecko. Opera has Presto. These engines are responsible for rendering HTML into a Document Object Model, executing JavaScript, and ultimately providing the layout of a Web page.

The same origin policy (SOP) is a fundamental security border with the browser. The abilities and visibility of content is restricted to the origin that initially loaded the content. Unlike a low-budget horror movie where demons can come from one origin to wreak havoc on another, JavaScript is supposed to be restricted to the origin from whence it came. JavaScript’s origin is the combination of at least the host name, port, and protocol of the containing page. In the age of mashups, this restriction is often considered an impediment to development. We’ll revisit SOP several times, beginning with Chapter 1, Cross-Site Scripting.

Background Knowledge

This book is far too short to cover ancillary topics in detail. Several attacks and countermeasures dip into subjects such as cryptography with references to hashes, salts, symmetric encryption, and random numbers. Other sections venture into ideas about data structures, encoding, and algorithms. Sprinkled elsewhere are references to regular expressions. Effort has been made to introduce these concepts with enough clarity to show how they relate to a situation. Some suggested reading has been provided where more background knowledge is necessary or useful. Hopefully, this book will lead to more curiosity on such topics. A good security practitioner will be conversant on these topics even if mathematical or theoretical details remain obscure.

The most important security tool for this book is the Web browser. Quite often, it’s the only tool necessary to attack a Web site. Web application exploits run the technical gamut of complex buffer overflows to single-character manipulations of the URI. The second most important tool in the Web security arsenal is a tool for sending raw HTTP requests. The following tools make excellent additions to the browser.

Netcat is the ancient ancestor of network security tools. It performs one basic function: open a network socket. The power of the command comes from the ability to send anything into the socket and capture the response. It is present by default on most Linux systems and MacOS X, often as the `nc` command. Its simplest use for Web security is as follows:

```bash
echo -e "GET / HTTP/1.0" | netcat -v mad.scientists.lab 80
```

Netcat has one failing for Web security tests: it doesn’t support SSL. Conveniently, the `OpenSSL` command provides the same functionality with only minor changes to the command line. An example follows.

```bash
echo -e "GET / HTTP/1.0" | openssl s_client -quiet -connect mad.scientists.lab:443
```

Local proxies provide a more user-friendly approach to Web security assessment than command line tools because they enable the user to interact with the Web site as usual with a browser, but also provide a way to monitor and modify the traffic between a
browser and a Web site. The command line serves well for automation, but the proxy is most useful for picking apart a Web site and understanding what goes on behind the scenes of a Web request. The following proxies have their own quirks and useful features.

- Burp Proxy (www.portswigger.net/proxy/)
- Fiddler (www.fiddler2.com/fiddler2/), only for Internet Explorer
- Paros (http://sourceforge.net/projects/paros/files/)
- Tamper Data (http://tamperdata.mozdev.org/), only for Firefox

**HOW THIS BOOK IS ORGANIZED**

This book contains seven chapters that address a serious type of attack against Web sites and browsers alike. Each chapter provides an example of how an attack has been used against real sites before exploring the details of how attackers exploit the vulnerability. The chapters do not need to be tackled in order. Many attacks are related or build on one another in ways that make certain countermeasures ineffective. That’s why it’s important to understand different aspects of Web security, especially the concept that security doesn’t end with the Web site, but extends to the browser as well.

**Chapter 1: Cross-Site Scripting**

Chapter 1 describes one of the most pervasive and easily exploited vulnerabilities that crop up in Web sites. XSS vulnerabilities are like the cockroaches of the Web, always lurking in unexpected corners of a site regardless of its size, popularity, or security team. This chapter shows how one of the most prolific vulnerabilities on the Web is exploited with nothing more than a browser and basic knowledge of HTML. It also shows how the tight coupling between the Web site and the Web browser can in fact be a fragile relationship in terms of security.

**Chapter 2: Cross-Site Request Forgery**

Chapter 2 continues the idea of vulnerabilities that target Web sites and Web browsers. CSRF attacks fool a victim’s browser into making requests that the user didn’t intend. These attacks are more subtle and difficult to block.

**Chapter 3: Structured Query Language Injection**

Chapter 3 turns the focus squarely onto the Web application and the database that drives it. SQL injection attacks are most commonly known as the source of credit-card theft. This chapter explains how many other exploits are possible with this simple vulnerability. It also shows that the countermeasures are relatively easy and simple to implement compared to the high impact successful attacks carry.
Chapter 4: Server Misconfiguration and Predictable Pages

Even the most securely coded Web site can be crippled by a poor configuration setting. This chapter explains how server administrators might make mistakes that expose the Web site to attack. This chapter also covers how the site’s developers might also leave footholds for attackers by creating areas of the site where security is based more on assumption and obscurity than well-thought-out measures.

Chapter 5: Breaking Authentication Schemes

Chapter 5 covers one of the oldest attacks in computer security: brute force and the login prompt. Yet brute force attacks aren’t the only way that a site’s authentication scheme falls apart. This chapter covers alternate attack vectors and the countermeasures that will – and will not – protect the site.

Chapter 6: Logic Attacks

Chapter 6 covers a more interesting type of attack that blurs the line between technical prowess and basic curiosity. Attacks that target a site’s business logic vary as much as Web sites do, but many have common techniques or target poor site designs in ways that can lead to direct financial gain for the attacker. This chapter talks about how the site is put together as a whole, how attackers try to find loopholes for their personal benefit, and what developers can do when faced with a problem that doesn’t have an easy programming checklist.

Chapter 7: Web of Distrust

Chapter 7 brings Web security back to the browser. It covers the ways in which malicious software, malware, has been growing as a threat on the Web. This chapter also describes ways that users can protect themselves when the site’s security is out of their hands.

WHERE TO GO FROM HERE

Hands-on practice provides some of the best methods for learning new security techniques or refining old ones. This book strives to provide examples and descriptions of the methodology for finding and preventing vulnerabilities. One of the best ways to reinforce this knowledge is by putting it to use against an actual Web application. It’s unethical and usually illegal to start blindly flailing away at a random Web site of your choice. That doesn’t limit the possibilities for practice. Scour sites such as SourceForge (www.sf.net/) for open-source Web applications. Download and install a few or a dozen. The act of deploying a Web site (and dealing with bugs in many of the applications) already builds experience with Web site concepts, programming patterns, and system administration that should be a foundation for
practicing security. Next, start looking for vulnerabilities in the application. Maybe it has an SQL injection problem or doesn’t filter user-supplied input to prevent XSS. Don’t always go for the latest release of a Web application; look for older versions that have bugs fixed in the latest version. You’ll also have the chance to deal with different technologies, from PHP to Java to C#, from databases such as MySQL to Postgresql to Microsoft SQL Server. Also, you’ll have access to the source code, so you can see why vulnerabilities arise, how a vulnerability may have been fixed between versions, or how you might fix the vulnerability. Hacking real applications (deployed in your own network) builds excellent experience.