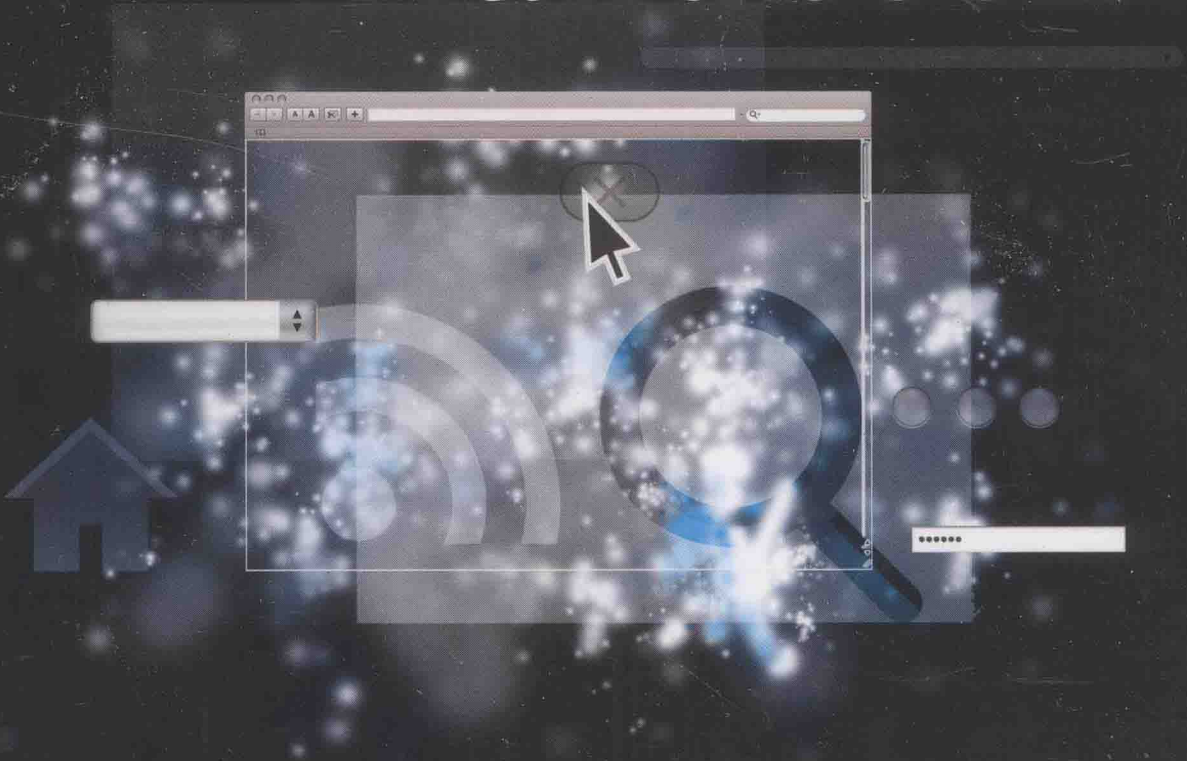
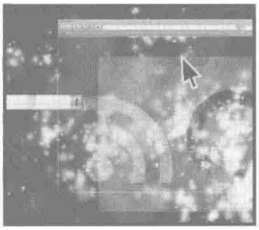
The Browser Hacker's Handbook



Wade Alcorn Christian Frichot Michele Orrù

WILEY



The Browser Hacker's Handbook

Wade Alcorn
Christian Frichot
Michele Orrù

WILEY

The Browser Hacker's Handbook

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

Published simultaneously in Canada

ISBN: 978-1-118-66209-0
ISBN: 978-1-118-66210-6 (ebk)
ISBN: 978-1-118-91435-9 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

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 publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2013958295

About the Authors

Wade Alcorn (@WadeAlcorn) has been in the IT security game for longer than he cares to remember. A childhood fascination with breaking stuff and solving puzzles put him on the path to his career.

Wade is the creator of BeEF (The Browser Exploitation Framework), which is considered one of the most popular tools for exploiting browsers. Wade is also the General Manager of the Asia Pacific arm of the NCC group, and has led security assessments targeting critical infrastructure, banks, retailers, and other enterprises.

Wade is committed to the betterment of IT security, and enjoys contributing to public groups and presenting at international conferences. He has published leading technical papers on emerging threats and has discovered vulnerabilities in widely used software.

Christian Frichot (@xntrik) has been into computers since the day his dad brought home an Amiga 1000. Having discovered it couldn't start Monkey Island with its measly 512KB of RAM, he promptly complained until the impressive 2MB extension was acquired. Since then, Christian has worked in a number of different IT industries, primarily Finance and Resources, until finally settling down to found Asterisk Information Security in Perth, Australia.

Christian is also actively involved in developing software; with a particular focus on data visualization, data analysis, and assisting businesses manage their security and processes more effectively. As one of the developers within the Browser Exploitation Framework (BeEF), he also spends time researching how to best leverage browsers and their technology to assist in penetration testing.

While not busting browsers, Christian also engages with the security community (have you seen how much he tweets?), not only as one of the Perth OWASP Chapter Leads, but also as an active participant within the wider security community in Perth.

Michele Orrù (@antisnatchor) is the lead core developer and "smart-minds-recruiter" for the BeEF project. He has a deep knowledge of programming in multiple languages and paradigms, and is excited to apply this knowledge while reading and hacking code written by others.

Michele loves lateral thinking, black metal, and the communist utopia (there is still hope!). He also enjoys speaking and drinking at a multitude of hacking conferences, including CONFidence, DeepSec, Hacktivity, SecurityByte, AthCon, HackPra, OWASP AppSec USA, 44Con, EUSecWest, Ruxcon, and more we just can't disclose.

Besides having a grim passion for hacking and programming, he enjoys leaving his Mac alone, while fishing on saltwater and "praying" for Kubrick's resurrection.

About the Contributing Authors

Ryan Linn (@sussurro) is a penetration tester, an author, a developer, and an educator. He comes from a systems administration and Web application development background, with many years of information technology (IT) security experience.

Ryan currently works as a full-time penetration tester and is a regular contributor to open source projects including Metasploit, BeEF, and the Ettercap project. He has spoken at numerous security conferences and events, including ISSA, DEF CON, SecTor, and Black Hat. As the twelfth step of his WoW addiction recovery program, he has gained numerous certifications, including the OSCE, GPEN, and GWAPT.

Martin Murfitt (@SystemSystemSyn) has a degree in physics but has worked as a penetration tester of various forms for all of his professional career since graduating in 2001 and stumbling randomly into the industry. Martin's passion for computing developed from a childhood of BBC micros in the 1980s. It isn't over yet.

Martin is a consultant and manager for the EMEA division of the global Trustwave SpiderLabs penetration testing team. SpiderLabs is the advanced security team at Trustwave responsible for incident response, penetration testing, and application security tests for Trustwave's clients.

Martin has discovered publicly documented vulnerabilities on occasion, presented sometimes or been working behind the scenes at conferences, such as Black Hat USA and Shmoocon, but generally prefers to be found contemplating.

About the Technical Editor

Dr.-Ing. Mario Heiderich (@0x6D6172696F) is founder of the German pen-test outfit Cure53, which focuses on HTML5, SVG security, scriptless attacks and—most importantly—browser security (or the abhorrent lack thereof). He also believes XSS can be eradicated someday (actually quite soon) by using JavaScript. Mario invoked the HTML5 security cheat sheet and several other security-related projects. In his remaining time he delivers training and security consultancy for larger German and international companies for sweet, sweet money and for the simple-minded fun in breaking things. Mario has spoken at a large variety of international conferences—both academic and industry-focused—co-authored two books and several academic papers, and doesn't see a problem in his two-year-old son having a tablet already.

Credits

Executive Editor
Carol Long

Project Editors
Ed Connor
Sydney Argenta Jones

Technical Editor
Mario Heiderich

Production Editor
Christine Mugnolo

Copy Editor
Kim Cofer

Editorial Manager
Mary Beth Wakefield

Freelancer Editorial Manager
Rosemarie Graham

Associate Director of Marketing
David Mayhew

Marketing Manager
Ashley Zurcher

Business Manager
Amy Knies

Vice President and
Executive Group Publisher
Richard Swadley

Associate Publisher
Jim Minatel

Project Coordinator, Cover
Todd Klemme

Compositor
Cody Gates,
Happenstance Type-O-Rama

Proofreaders
Josh Chase and Sarah Kaikini,
Word One New York

Indexer
Johnna VanHoose Dinse

Cover Designer and Image
© Wiley

Acknowledgments

Nothing worthwhile in my life could be achieved without two very important people. A huge thank you to my beautiful wife, Carla, for her inexhaustible support and immeasurable inspiration. Though she is not mentioned on the cover, her hand has been involved in refining every word of this book. I also owe much to my hero and son, Owen. Without him continually showing that every life challenge is best confronted with a grin firmly planted from ear to ear, all obstacles would be so much greater.

I have also been lucky enough to work almost a decade with Rob Horton and Sherief Hammad. They have always been a source of continual encouragement, and have provided a supportive workplace that fostered creativity and lateral thinking. And of course, thanks to Michele and Christian for taking this literary journey with me.

— Wade Alcorn

I first met her while breaking systems in a bank, and without her unending patience I would not have been able to help write this book. To my wonderful wife Tenille, I thank you with all my heart, and to our daughter growing inside you—this book is for you (make sure you practice responsible hacking little one). I must also thank the rest of my family, to my mother Julia and father Maurice for providing me all the opportunities in life that have allowed me to participate in this amazing information security industry. To my sisters Hélène, Justine and Amy, you guys are inspiring, and your support has been very much appreciated. To my Asterisk Info Sec family, for letting me complain about how flipping hard this was, and for giving me the time to contribute to this book, thank you so much David Taylor, Steve Schupp, Cole Bergersen, Greg Roberts and Jarrod Burns. I must also thank all of the Australian and New Zealand

hacker security crowd, all the friends that I've gotten to know over the Internet and at conferences, I love being part of this community with you guys, keep on rocking. And of course Wade and Michele, I have to thank you guys for inviting me into this monumental task, for your patience, for everything you've taught me, and for putting up with my crap!

— Christian Frichot

First of all I would like to thank my beloved Ewa for the moral support during the endless days and nights I've spent doing research and working on this book. Great devotion goes to my parents who always supported me and gave me the possibility to study and learn new things. Huge thanks to my good friends Wade Alcorn and Mario Heiderich for research inspiration and mind-blowing discussions. Without them this book wouldn't have reached the quality we were aiming for. Cheers to everyone who believed and still believes in Full Disclosure as the way bugs should be disclosed. Finally, but not lastly, a big hug to all my hacking friends and security researchers (you know who you are), who have shared with me exploits and conference hangovers.

— Michele Orrù

This book is the result of a team effort. First and foremost, we would like to acknowledge and thank our two contributing authors, Ryan Linn and Martin Murfitt. We are also indebted to the wider security community, particularly the cast of many who have contributed to BeEF over the years. Much of their effort has provided the foundation for what is presented in this book today.

The good people at Wiley and the book's Technical Editor are also due a very large thank you. Mario Heiderich, Carol Long, and Ed Connor must have special mention for their (unending) patience, support, and expertise.

Thanks to Krzysztof Kotowicz, Nick Freeman, Patroklos Argyroudis, and Chariton Karamitas for their expert contributions. Though we can't thank everyone individually, there are some that we would like to give a special mention. They are: Brendan Coles, Heather Pilkington, Giovanni Cattani, Tim Dillon, Bernardo Damele, Bart Leppens, George Nicolau, Eldar Marcussen, Oliver Reeves, Jean-Louis Huynen, Frederik Braun, David Taylor, Richard Brown, Roberto Suggi Liverani, and Ty Miller. Undoubtedly we have missed important people. If we have, the error is by omission, not intention.

— From all of us

Introduction

Overview of This Book

You have chosen to read a book that will provide you with a practical understanding of hacking the everyday web browser and using it as a beachhead to launch further attacks. The attacks will focus on the most popular browsers and occasionally delve into the less mainstream ones. You will largely explore Firefox, Chrome, and Internet Explorer. You will even dip your toes into the water of modern mobile browsers and, although these won't be the primary focus, a lot of the attacks are relevant to them also.

Attackers and defenders both need to understand the dangers the web browser has opened up for users. The reason is obvious. The web browser is possibly the most important piece of software so far this century. It is humanity's most popular gateway to access the online environment—so much so that you have watched it grow from cumbersome desktop software to a dominant application on your phone, gaming console, and even your humble TV. It is today's Swiss Army knife of presenting, retrieving, and navigating data. Since Sir Tim Berners-Lee invented his "little web browser that could" in 1990, this overachieving application has become one of the most recognizable pieces of software in the world.

Various estimates are being thrown about regarding the number of people globally using web browsers. Doing some "back of the napkin" calculations will reveal some extraordinary numbers. If you say that about one-third of the global population is using the Internet, then you could estimate about 2.3 billion browsers. Drawing further assumptions, you may discover that some are using n+1 browsers. Some are using a browser at home, at work, and on their phones. Even without Stephen Hawking's mathematical insights, you have probably arrived at a stupendous number.

Given this astonishing number of web browsers, it is not surprising that with this popularity comes a plethora of security issues and opportunities for exploitation. Written from the perspective of the hacker, this book will teach you how to hack, and thereby how to defend, the modern browser in all its glory.

Who Should Read This Book

Do you have a technical background and an interest in understanding the practical risks of web browsers? If yes, then this book is for you. You may be looking to defend your infrastructure or attack your client's assets. You may have a role as an administrator, developer, or even an information security professional. Like a lot of us, you may simply have an overwhelming passion for security and are continually looking to augment your knowledge.

This book has been written assuming you use a web browser regularly and have had cause to look under the hood on occasion. It will be beneficial for you to already have a grasp of fundamental security concepts or be happy to invest a little time in some background research. The concept of the server-client model, the HTTP protocol, and general security concepts should not be new to you.

Although it isn't essential to have a programming background, it would be useful to have some basic knowledge of the principles when reviewing the code snippets. Numerous examples and demonstrations are provided throughout the book to give you hands-on experience. These are written in various languages with an emphasis on JavaScript, due to its dominance within browsers. As unlikely as it may be, if you haven't used JavaScript before, don't be concerned. The code also comes with explanations.

How This Book Is Organized

This book contains 10 chapters that are broadly categorized based on the attacking method. Where possible, sections are divided into vulnerability classes, but this is not strictly the case. The book has been organized in a structure that the authors envisage may be helpful to you as you embark upon a professional security engagement.

During any security engagement, it is unlikely you'll follow this book from cover to cover. Rather, you will hop from one chapter to another, starting from the introductory chapters and then branching into the most relevant chapter. Alternatively, you may leap into a section where a concept is discussed in detail. To support this more dynamic usage of the book, some concepts are replicated to add context and coherence to the individual topics.

Each chapter concludes with a set of questions for you to ponder. These questions will provide you with an opportunity to consolidate your understanding of the core concepts of the chapter.

Chapter 1: Web Browser Security

This chapter starts you on your browser hacking journey. Your first step is to explore important browser concepts and some of the core problems with browser security. You explore the *micro perimeter* paradigm needed to defend organizations today, and ponder some fallacies that continue to propagate insecure practices.

This chapter also examines a methodology specifying how attacks employing the browser can be launched. It covers the attack surface presented by the browser and how it increases the exposure of assets previously assumed protected.

Chapter 2: Initiating Control

Every single time a web browser connects to the web, it is asking for instructions. The browser then dutifully carries out the orders it has been provided by the web server. Needless to say, boundaries do exist, but the browser provides a powerful environment for attackers to employ.

This chapter walks you through the first phase of browser attacks by exploring how to execute your code within the target browser. You sample the delights of Cross-site Scripting vulnerabilities, Man-in-the-Middle attacks, social engineering, and more.

Chapter 3: Retaining Control

The initiation techniques discussed up to this point only allow you to execute your instructions once. This chapter introduces how to maintain communication and persistence, giving you interactive control with the ability to execute multiple rounds of commands.

In a typical hacking session, you will want to maintain a communication channel with the browser and, where possible, persist your control across restarts. Without this, you will quickly find yourself back at square one trying to entice your target to connect over and over again.

In this chapter, you learn how to use a payload to maintain communication with the browser, enabling you to send multiple iterations of instructions. This will ensure that you don't waste any opportunities once you have received that all-important initial connection. Armed with this knowledge, you are now ready to launch the various attacks presented in the following chapters.

Chapter 4: Bypassing the Same Origin Policy

In very basic terms, the Same Origin Policy (SOP) restricts one website from interacting with another one. It is possibly the most fundamental concept in web browser security. You would, therefore, expect that it would be consistent across browser components and trivial to predict the impacts of common actions. This chapter shows you that this is not the case.

Web developers are poked with an SOP stick at almost every turn; there is variance between how SOP is applied to the browser itself, extensions, and even plugins. This lack of consistency and understanding provides attackers opportunities to exploit edge cases.

This chapter explores bypassing the different SOP controls in the browser. You even discover issues with drag-and-drop and various UI redressing and timing attacks. One of the more surprising things you learn in this chapter is that with the right coding, SOP bypasses can transform the browser into an HTTP proxy.

Chapter 5: Attacking Users

Humans are often referred to as the weakest link in security. This chapter focuses on attacks targeting the unsuspecting user's wetware. Some of the attacks further leverage social engineering tactics discussed in Chapter 2. Other attacks exploit *features* of browsers, and their trust in received code.

In this chapter, you explore de-anonymization and covertly enabling the web camera, as well as running malicious executables with and without any explicit user intervention.

Chapter 6: Attacking Browsers

While this entire book is about attacking the browser and circumventing its security controls, this chapter focuses on what could be referred to as the *bare-bones* browser. That is, the browser without the extensions and plugins.

In this chapter, you explore the process of directly attacking the browser. You delve into fingerprinting the browser to distinguish between vendors and versions. You also learn how to launch attacks and compromise the machine running the browser.

Chapter 7: Attacking Extensions

This chapter focuses on exploiting vulnerabilities in browser extensions. An extension is software that adds (or removes) functionality to (or from) the web browser. An extension is not a standalone program unlike their second cousins, plugins. You might be familiar with extensions like LastPass, Firebug, AdBlock, and NoScript.

Extensions execute code in trusted zones with increased privileges and take input from less trusted zones like the Internet. This will ring alarm bells for seasoned security professionals. There is a real risk of injection attacks, and in practice, some of these attacks lead to remote code execution.

In this chapter, you explore the anatomy of extension attacks. You delve into privilege escalation exploits that will give you access to the privileged browser (or `chrome://`) zone and result in command execution.

Chapter 8: Attacking Plugins

This chapter focuses on attacking web browser plugins, which are pieces of software that add specific functionality to web browsers. In most instances, plugin software can run independently without the web browser.

Popular plugins include Acrobat Reader, Flash Player, Java, QuickTime, RealPlayer, Shockwave, and Windows Media Player. Some of these are necessary for your browsing experience, and some for your business functions. Flash is needed for sites like YouTube (which is potentially moving to HTML5) and Java is required for business functions such as WebEx.

Plugins have been plagued with vulnerabilities and continue to be a rich source of exploits. As you'll discover, plugin vulnerabilities remain one of the most reliable avenues to take control of a browser.

In this chapter, you explore analyzing and exploiting browser plugins using popular, freely available tools. You learn about bypassing protection mechanisms like Click to Play and taking control of the target through vulnerabilities in the plugins.

Chapter 9: Attacking Web Applications

Your everyday web browser can conduct powerful web-based attacks while still abiding by accepted security controls. Web browsers are designed to communicate to web servers using HTTP. These HTTP functions can be turned against themselves to achieve a compromise of a target that is not even on the current origin.

This chapter focuses on attacks that can be launched from the browser without violating the SOP. You learn various tricks that allow cross-origin fingerprinting of resources and even cross-origin identification of common web application vulnerabilities. You may be surprised to learn that when using the browser, it is possible to discover and exploit cross-origin Cross-site Scripting and SQL injection vulnerabilities, too.

By chapter's end, you'll understand how to achieve cross-origin remote code execution. You will also discover Cross-site Request Forgery attacks, time-based delay enumeration, attacking authentication, and Denial-of-Service attacks.

Chapter 10: Attacking Networks

This final attacking chapter covers identifying the intranet's attack surface by port scanning to discover previously unknown hosts. The exploration continues by presenting techniques such as NAT Pinning.

In this chapter, you also discover attacks that use the web browser to communicate directly to non-web services. You learn how to harness the power of the Interprotocol Exploitation technique to compromise targets on the browser's intranet.

Epilogue: Final Thoughts

By this stage in the book you will have learned numerous offensive techniques and the chapters should now serve as a reference to quickly re-ramp up your knowledge. We leave you with some thoughts to ponder, particularly around the future of browser security.

What's on the Web

The website that accompanies this book is located at `https://browserhacker.com` or the Wiley website at: `www.wiley.com/go/browserhackershandbook`. On this site you will find information that augments the contents of this book. It is not a substitute, but the details will complement the knowledge you get from within the chapters.

The website also includes code snippets for you to copy and paste. This will save you from having to transcribe them manually and has the added benefit of (hopefully) delaying the onset of RSI! You'll also find demonstration videos to view and answers to each chapter's questions for you to check your knowledge.

Our modesty requires us to admit that there will inevitably be mistakes in this book. It is an unfortunate truth that all but one of the authors of this book is fallible (we are still in violent disagreement about which one of us is the infallible one). Please check `https://browserhacker.com` to find out if we have determined the fallible one and, of course, for the corrections to mistakes discovered by our readers. If you find an error, please check the site and, if it isn't listed, kindly notify us.

Compiling Your Arsenal

This book covers various tools you can employ to hack web browsers and it is valuable to have a variety in your toolkit.

An important point to stress is that this book aims to give you knowledge of how the tools work from a low level. This will be an extremely valuable insight as your skill level increases. The aim is not only to teach you how to use tools, but to *understand* them and enable you to spot the inevitable false positives.

It is hoped that you will take an understanding that all tools have weaknesses and that you should combine your knowledge with this fact in your security engagements. The most important tool in your toolkit is your knowledge. The authors' primary aim is to expand your understanding and not your software library.

A couple of the tools you will see frequently throughout this book are the Browser Exploitation Framework (BeEF) and Metasploit. Of course, many others are covered and you will become familiar with all their strengths and weaknesses.

The authors are core developers on the BeEF project and steered the development of this community tool to match the methodology described herein. Numerous examples have come from the BeEF codebase where the majority of the processes have been automated.

Authorization Denied

This is a good point to pause in the book and highlight the professionalism needed within the security disciplines. In no way should anything in this book be interpreted as providing permission or encouragement to conduct an illegal act.

Ensure that you have received full permission prior to conducting a hacking engagement. This is true of most of the security disciplines and is applicable for all the techniques discussed in this book.

Good to Go!

Web browser security is one of the fastest moving arms races on the Internet. This makes it a fascinating and fun area for anyone interested in security to get involved. The pace is not slowing because businesses continually push the boundaries of what browsers can do.

We have seen large and small companies alike aggressively changing the assumption that usable and responsive software runs solely on the desktop computer. Anyone predicting a decline in browser popularity should double-check their Ouija board because they probably still have that buggy Java plugin enabled!

Combine the arms race and business interests with the continually changing web browser attack surface, and the security challenges won't stop coming. So, let's jump right in and start hacking browsers!

Contents