SYNGRESS

A Guide to Kernel Exploitation

Attacking the Core

Enrico Perla
Massimiliano Oldani

A Guide to Kernel Exploitation

Attacking the Core

Enrico Perla

Massimiliano Oldani

Technical Editor

Graham Speake

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Syngress is an imprint of Elsevier

SYNGRESS®

Acquiring Editor: Rachel Roumeliotis
Development Editor: Matthew Cater
Project Manager: Julie Ochs
Designer: Alisa Andreola

Syngress is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods or professional practices, may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information or methods described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data

Perla, Enrico.
A guide to kernel exploitation : attacking the core / Enrico Perla, Massimiliano Oldani.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-59749-486-1 (pbk. : alk. paper)
1. Operating systems (Computers)—Security measures. 2. Computer security. I. Massimiliano, Oldani. II. Title.
QA76.76.O63P5168 2010
005.8—dc22 2010027939

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

Printed in the United States of America

10 11 12 13 14 10 9 8 7 6 5 4 3 2 1

Typeset by: diacriTech, Chennai, India

A Guide to Kernel Exploitation

Foreword

When I was originally asked to write a Foreword for this book, I refused because I didn't want to show up in the light dedicated to others whose hard work resulted in the book you hold in your hands. However, after proofreading some of the book's chapters I realized that it would be sad to miss the opportunity, and that it is a great honor to write a few words in a book authored by two of the world's best kernel exploit developers.

I rarely read books about exploitation techniques because they usually provide little or outdated knowledge or simply enumerate exploits done by others. Additionally, books cannot provide the learning effect of hands-on exploit development or the fun of a '#' prompt after days of hard work, especially if a kernel vulnerability is exploited. It's about time that someone transformed this feeling into paper with the benefit of saving other developers time, a lot of crashes, and headaches.

Besides all the nice tricks and exploitation martial arts, writing exploits, and kernel exploits in particular, is engineering that requires a deep understanding of operating system fundamentals. This book is definitely helpful for such purposes and fills the gap between all the kernel and driver programming books on my bookshelf.

I know for sure who around the world will read this book, and I hope that a lot of kernel and driver developers are among that readership. My next kernel code review job will definitely come, and I hope my printed copy of this book arrives before it does.

Sebastian Krahmer
System programmer and exploit engineer

Preface

INFORMATION IN THIS SECTION

- Book Overview
- How This Book Is Organized

BOOK OVERVIEW

With the number of security countermeasures against user-land exploitation greater than ever these days, kernel-level exploitation is becoming increasingly popular among attackers and, generically, exploit writers. Playing with the heart of a computer's operating system can be a dangerous game. This book covers the theoretical techniques and approaches needed to develop reliable and effective kernel-level exploits and applies them to different operating systems—namely, UNIX derivatives, Mac OS X, and Windows.

Kernel exploits require both art and science to achieve. Every OS has its quirks, so every exploit must be molded to take full advantage of its target. This book discusses the most popular OS families—UNIX derivatives, Mac OS X, and Windows—and how to gain complete control over them.

Concepts and tactics are presented categorically so that even when a specifically detailed vulnerability has been patched, the foundational information that you have read will help you to write a newer, better attack if you are a hacker; or a more concrete design and defensive structure if you are a pen tester, auditor, or the like.

HOW THIS BOOK IS ORGANIZED

This book is divided into four parts and nine chapters. **Part I, A Journey to Kernel Land**, introduces our target, the kernel, and aims at setting down the theoretical basis on which we will build throughout the rest of the book. Here's what you'll find in this part of the book:

- **Chapter 1, From User-Land to Kernel-Land Attacks**, introduces the world of exploitation and analyzes what has caused security researchers and attackers to change their focus from targeting user-land applications to exploiting the core of a running system, the kernel.
- **Chapter 2, A Taxonomy of Kernel Vulnerabilities**, builds a classification of different types of vulnerabilities (bug classes), looking at common traits and exploitation approaches. The more we can model different bug classes, the better we can design and invent reliable and effective techniques. This classification is also handy when we look at the problem from the other side

of the fence: defense. The more we understand about bug classes, the better we can invent protections and countermeasures against them.
- **Chapter 3, Stairway to Successful Kernel Exploitation**, dissects the building blocks of an exploit and describes techniques and best approaches for each bug class presented in Chapter 2. Although operating systems differ in the way they implement their subsystems, this chapter aims to provide approaches that are easily applicable to different kernels as well as different architectures.

Part II, The UNIX Family, Mac OS X, and Windows, is where we start getting our hands dirty, delving deep into the details regarding different operating systems and writing exploits for them that target various bug classes. For each operating system, we also spend time covering debugging tools and approaches, which become extremely useful when writing exploits. Where possible, we present exploits for "real" vulnerabilities rather than crafted examples. Here's what you'll find in this part of the book:

- **Chapter 4, The UNIX Family**, analyzes UNIX derivative systems, focusing largely on Linux and somewhat on the (Open)Solaris operating systems. A part of the chapter is also dedicated to debugging techniques with the main tools these operating systems offer (dynamic tracing, in-kernel debugger, etc.).
- **Chapter 5, Mac OS X**, covers the Leopard version of the increasingly popular Mac OS X operating system. Along with an analysis of the main bug classes (e.g., stack and heap exploitation), we present an analysis of how the closed parts of the kernel can be reverse engineered when looking for vulnerabilities.
- **Chapter 6, Windows**, covers the most popular operating system in the world, Microsoft Windows. Unlike the preceding chapters, in this chapter we do not have the sources of the kernel; rather, our understanding of the internals (and vulnerabilities/exploitation approaches) comes from reverse engineering the various kernel parts. Even more so than in Chapters 4 and 5, learning about the debugging and reverse-engineering tools is important here, and we dedicate a part of the chapter to this topic.

Part III, Remote Kernel Exploitation, moves our attention from the local scenario (the one that is common for kernel attacks) to the remote case. Indeed, we enter trickier territory, where many of the techniques we have learned to use in local attacks are simply no longer applicable. Although bug classes remain the same, we need to add a new set of weapons to our arsenal. Part III is divided into two chapters, harking back to the structure of the previous part of the book (Part I being more theoretical and Part II being more practical). Here's what you'll find in this part of the book:

- **Chapter 7, Facing the Challenges of Remote Kernel Exploitation**, starts with the theory, analyzing why and how much the remote scenario affects our approaches and presenting new techniques to target remote issues. Despite this chapter being a "theoretical" chapter, a few practical examples are presented, in particular focusing on the Windows operating system, since the UNIX (Linux) case gets an entire chapter (the following one) dedicated to it.

- **Chapter 8, Putting It All Together: A Linux Case Study**, is a step-by-step analysis of the development of a reliable, one-shot, remote exploit for a real vulnerability—a bug affecting the SCTP subsystem (http://cve.mitre.org/cgi-bi/cvename.cgi?name=CVE-2009-0065) found in the Linux kernel.

Part IV, Final Words, concludes the book, wrapping up our analysis of kernel (in)security. It is composed of a single chapter:

- **Chapter 9, Kernel Evolution: Future Forms of Attack and Defense**, where we build on what we have learned about kernel exploitation and look at what the future may hold. To be able to put some order to the many aspects of attack and defense techniques, in this chapter we turn to the basics of computer security: information flow control. We then use it as our looking glass to inspect and understand some fundamental traits of bugs and exploits so that we can better understand where the future will take them.

The source code for all the exploits and tools presented in this book is available on the book's Web site, www.attackingthecore.com, which is also the main point of reference to report errors; to look for extra material; and, if you wish, to contact us.

Please be advised that the superscripted numbers in the text indicate corresponding numbered entries in the section entitled Endnotes at the end of chapters. Footnotes in this book use a superscripted, lettered format.

CONCLUSION

Writing a book is a fantastic yet terrifying experience. It is a chance for an author to document the many concepts that have been floating through his or her mind regarding his or her favorite topic. Writing this book was a challenge for us, on many levels. We strived to be clear and correct in the explanation, transfer the passion (and fun) that is involved in finding ways to break things (or prevent the breakage), and offer information that is valuable not only when the book is printed, but also for some time thereafter. We hope you'll like this effort as much as we have enjoyed putting it together for you.

Acknowledgments

This book is dedicated to all those that still believe that when it comes to security, your ability with your code editor (and shell) is more important than your ability with your mail client.

Various people helped, supported, and patiently nurtured this manuscript through to a final product. Simply stated, without them, what you are holding in your hands right now (or checking through your favorite PDF reader) would not have been possible. We would like in particular to thank:

- Matthew Cater, Rachel Roumeliotis, Graham Speake, Audrey Doyle, and Julie Ochs for putting up (more than once) with a dancing schedule and our constant requests to increase the number of pages from the original estimate.
- Nemo for his amazing material for Chapter 5 and the constant feedback.
- Ruggiero Piazzolla, for helping with the website and especially, for making it easy on the eyes.
- Marco Desiati and Michele Mastrosimone for helping with the art. Our original attempts looked like childish sketches compared to their final results.
- Abh for tirelessly spending lots of his time proofreading, reviewing, and improving the contents and code examples contained in this book.
- Sebastian Krahmer for contributing the Foreword, reviewing many of the chapters, and for the endless discussions about techniques and ideas.
- (In random order) Andrea Lelli, Scott Rotondo, *xorl* (nice blog, btw!), Brad Spengler, Window Snyder, Julien Vanegue, Josh Hall, Ryan Austin, Bas Albert, Igor Falcomata', *clint*, Reina Alessandro, Giorgio Fedon, Matteo Meucci, Stefano Di Paola, Antonio Parata, Francesco Perna, Alfredo Pesoli, Gilad Bakas, David Jacoby, and Ceresoni Andrea for sending feedback and ideas about the book and helping to improve its overall quality (and, occasionally, providing a bed or a couch to crash on). We are sure we have forgotten others here (never has the sentence "*you* know who you are" been more appropriate)...sorry about that.

Last but not least, there are a few special thanks missing, but they are personal, rather than shared.

Enrico would like to thank Mike Pogue and Jan Setje-Eilers for, well, just about everything they have done and Lalla, Franco, and Michela for being a fantastic family. A special thanks goes to the 9:00 a.m. and 10:30 p.m. phone calls, which have made living (thousands of) miles away from home much, much closer to Home.

Massimiliano would like to give the following thanks:

- To *halfdead* for making me see that it is still possible to have a lot of fun with the fantastic security world.
- To my wonderful family: Noemi, Manuela, Giuseppe, Stefano (Bruce), and especially Irene, who gave up a lot of weekends to support me during all the months spent writing this book; I really love you.

About the Authors

Enrico Perla currently works as a kernel programmer at Oracle. He received his B.Sc/ in Computer Science from the University of Torino in 2007 and his M.Sc. in Computer Science from Trinity College Dublin in 2008. His interests range from low-level system programming to low-level system attacking, exploiting, and exploit countermeasures.

Massimiliano Oldani currently works as a Security Consultant at Emaze Networks. His main research topics include operating system security and kernel vulnerabilities.

About the Technical Editor

Graham Speake (CISSP #56073, M.Inst. ISP) is a Principal Systems Architect at Yokogawa Electric Corporation, a major industrial automation supplier. He currently provides security advice and solutions to internal developers and customers in many countries. His specialties include industrial automation and process control security, penetration testing, network security, and network design. Graham is a frequent speaker at security conferences and often presents security training to customers around the world. Graham's background includes positions as a security consultant at both BP and ATOS/Origin and as an engineer at the Ford Motor Company.

Graham holds a bachelor's degree from the Swansea University in Wales and is a member of the ISA. Graham was born in the United Kingdom, but now lives in Houston, Texas, with his wife, Lorraine and daughter, Dani.

Contents

PART I A JOURNEY TO KERNEL LAND

PART III REMOTE KERNEL EXPLOITATION

PART IV FINAL WORDS