INTRODUCTION

We came up with the idea for this book when, having published a series of articles and blog posts about rootkits and bootkits, we realized the topic wasn't getting nearly as much attention as it deserved. We felt there was a bigger picture, and we wanted a book that tried to make sense of it all—one that generalized the medley of nifty tricks, operating system architectural observations, and design patterns used by attacker and defender innovations. We looked for such a book and found none, so we set out to write the one we wanted to read.

It took us four and a half years, longer than we planned and, regrettably, much longer than we could count on for the prospective readers and supporters of the early access editions to stay with us. If you are one of these early access supporters and are still reading this book, we're humbled by your continued devotion!

During this time, we observed the coevolution of offense and defense. In particular, we saw Microsoft Windows defenses dead-ending several major branches of rootkit and bootkit designs. You'll find that story in the pages of this book.

We also saw the emergence of new classes of malware that target the BIOS and the chipset firmware, beyond the reach of current Windows defensive software. We'll explain how this coevolution developed and where we expect its next steps to take us.

Another theme of this book is the development of the reverse engineering techniques targeting the early stages of the OS boot process. Traditionally, the earlier in the long chain of the PC boot process a piece of code came into play, the less observable it was. This lack of observability has long been confused with security. Yet, as we dig into the forensics of bootkits and BIOS implants subverting low-level operating system technologies such as Secure Boot, we see that security by obscurity fares no better here than in other areas of computer science. After a short time (which is only getting shorter on the internet time scale), the security-by-obscurity approach comes to favor the attackers more than the defenders. This idea has not been sufficiently covered in other books on the subject, so we try to fill this gap.

Why Read This Book?

We write for a very broad circle of information security researchers interested in how advanced persistent malware threats bypass OS-level security. We focus on how these advanced threats can be observed, reverse engineered, and effectively analyzed. Each part of the book reflects a new stage of the evolutionary development of advanced threats, from their emergence as narrow proofs of concept, to their subsequent spread among threat actors, and finally to their adoption into the sneakier arsenal of targeted attacks.

However, we aim to reach a wider audience than just PC malware analysts. In particular, we hope that embedded systems developers and cloud security specialists will find this book equally useful, considering that the threat of rootkits and other implants looms large in their respective ecosystems.

What's in the Book?

We start with an exploration of rootkits in Part 1, where we introduce the internals of the Windows kernel that historically served as the rootkits' playground. Then in Part 2, we shift focus toward the OS boot process and the bootkits that developed after Windows started hardening its kernel mode. We dissect the stages of the boot process from the attacker's perspective, paying particular attention to the new UEFI firmware schemes and their vulnerabilities. Finally, in Part 3, we focus on the forensics of both the classic OS rootkit attacks and newer bootkit attacks on the BIOS and firmware.

Part 1: Rootkits

This part focuses on the classic OS-level rootkits during their heyday. These historic rootkit examples provide valuable insights into how attackers see the operating system internals and find ways to reliably compose their implants into them, using the OS's own structure.

Chapter 1: What's in a Rootkit: The TDL3 Case Study We start exploring how rootkits work by telling the story of one of the most interesting rootkits of its time, based on our own encounters with its diverse variants and our analysis of these threats.

Chapter 2: Festi Rootkit: The Most Advanced Spam and DDoS Bot Here we analyze the remarkable Festi rootkit, which used the most advanced stealth techniques of its time to deliver spam and DDoS attacks. These techniques included bringing along its own custom kernel-level TCP/IP stack.

Chapter 3: Observing Rootkit Infections This chapter takes our journey into the depths of the operating system kernel, highlighting the tricks attackers used to fight for control of the kernel's deeper layers, such as intercepting system events and calls.

Part 2: Bootkits

The second part shifts focus to the evolution of bootkits, the conditions that spurred that evolution, and the techniques for reverse engineering these threats. We'll see how bootkits developed to implant themselves into the BIOS and exploit UEFI firmware vulnerabilities.

Chapter 4: Evolution of the Bootkit This chapter takes a deep dive into the (co)evolutionary forces that brought bootkits into being and guided their development. We'll look at some of the first bootkits discovered, like the notorious Elk Cloner.

Chapter 5: Operating System Boot Process Essentials Here we cover the internals of the Windows boot process and how they've changed over time. We'll dig into specifics like the Master Boot Record, partition tables, configuration data, and the *bootmgr* module.

Chapter 6: Boot Process Security This chapter takes you on a guided tour of Windows boot process defense technologies, such as Early Launch Anti-Malware (ELAM) modules, the Kernel-Mode Code Signing Policy and its vulnerabilities, and newer virtualization-based security.

Chapter 7: Bootkit Infection Techniques In this chapter, we dissect the methods of infecting boot sectors and look at how these methods had to evolve over time. We'll use some familiar bootkits as examples: TDL4, Gapz, and Rovnix.

Chapter 8: Static Analysis of a Bootkit Using IDA Pro This chapter covers the methods and instruments for static analysis of bootkit infections. We'll guide you through the analysis of the TDL4 bootkit as an example, and we'll provide materials for you to use in your own analysis, including a disk image to download.

Chapter 9: Bootkit Dynamic Analysis: Emulation and Virtualization Here we shift focus to dynamic analysis methods, using the Bochs emulator and VMware's built-in GDB debugger. Again, we'll take you through the steps of dynamically analyzing the MBR and VBR bootkits.

Chapter 10: An Evolution of MBR and VBR Infection Techniques: Olmasco This chapter traces the evolution of the stealth techniques used to take bootkits into the lower levels of the boot process. We'll use Olmasco as an example, looking at its infection and persistence techniques, the malware functionality, and payload injection.

Chapter 11: IPL Bootkits: Rovnix and Carberp Here we take a look under the hood of two of the most complex bootkits, Rovnix and Carberp, which targeted electronic banking. These were the first bootkits to target the IPL and evade contemporary defense software. We'll use VMware and IDA Pro to analyze them.

Chapter 12: Gapz: Advanced VBR Infection We'll demystify the pinnacle of the bootkit stealth evolution: the mysterious Gapz rootkit, which used the most advanced techniques of its time to target the VBR.

Chapter 13: Rise of MBR Ransomware In this chapter, we look at how bootkits rebounded in ransomware threats.

Chapter 14: UEFI Boot vs. the MBR/VBR Boot Process Here we explore the boot process of UEFI BIOS designs—essential information for discovering the newest malware evolutions.

Chapter 15: Contemporary UEFI Bootkits This chapter covers our original research into the various BIOS implants, both proofs of concept and those deployed in the wild. We'll discuss methods for infecting and persisting on the UEFI BIOS and look at UEFI malware found in the wild, like Computrace.

Chapter 16: UEFI Firmware Vulnerabilities Here we take an in-depth look at different classes of modern BIOS vulnerabilities that enable the introduction of BIOS implants. This is a deep exploration of UEFI vulnerabilities and exploits, including case studies.

Part 3: Defense and Forensic Techniques

The final part of the book addresses the forensics of bootkits, rootkits, and other BIOS threats.

Chapter 17: How UEFI Secure Boot Works This chapter takes a deep dive into the workings of the Secure Boot technology and its evolution, vulnerabilities, and effectiveness.

Chapter 18: Approaches to Analyzing Hidden Filesystems This chapter provides an overview of the hidden filesystems used by malware and methods of detecting them. We'll parse a hidden filesystem image and introduce a tool we devised: the HiddenFsReader.

Chapter 19: BIOS/UEFI Forensics: Firmware Acquisition and Analysis Approaches This final chapter discusses approaches to detecting the most advanced state-of-the-art threats. We look at hardware, firmware, and software approaches, using various open source tools, like UEFITool and Chipsec.

How to Read This Book

All the specimens of threats discussed in the book, as well as other supporting materials, can be found at the book's website, *https://nostarch.com/rootkits/*. This site also points to the tools used in the bootkits' analysis, such as the source code of the IDA Pro plug-ins that we used in our original research.