INTRODUCTION

The phone rings, and the networking guys tell you that you've been hacked and that your customers' sensitive information is being stolen from your network. You begin your investigation by checking your logs to identify the hosts involved. You scan the hosts with antivirus software to find the malicious program, and catch a lucky break when it detects a trojan horse named *TROJ.snapAK*. You delete the file in an attempt to clean things up, and you use network capture to create an intrusion detection system (IDS) signature to make sure no other machines are infected. Then you patch the hole that you think the attackers used to break in to ensure that it doesn't happen again.

Then, several days later, the networking guys are back, telling you that sensitive data is being stolen from your network. It seems like the same attack, but you have no idea what to do. Clearly, your IDS signature failed, because more machines are infected, and your antivirus software isn't providing enough protection to isolate the threat. Now upper management demands an explanation of what happened, and all you can tell them about the malware is that it was *TROJ.snapAK*. You don't have the answers to the most important questions, and you're looking kind of lame.

How do you determine exactly what *TROJ.snapAK* does so you can eliminate the threat? How do you write a more effective network signature? How can you find out if any other machines are infected with this malware? How can you make sure you've deleted the entire malware package and not just one part of it? How can you answer management's questions about what the malicious program does?

All you can do is tell your boss that you need to hire expensive outside consultants because you can't protect your own network. That's not really the best way to keep your job secure.

Ah, but fortunately, you were smart enough to pick up a copy of *Practical Malware Analysis*. The skills you'll learn in this book will teach you how to answer those hard questions and show you how to protect your network from malware.

What Is Malware Analysis?

Malicious software, or *malware*, plays a part in most computer intrusion and security incidents. Any software that does something that causes harm to a user, computer, or network can be considered malware, including viruses, trojan horses, worms, rootkits, scareware, and spyware. While the various malware incarnations do all sorts of different things (as you'll see throughout this book), as malware analysts, we have a core set of tools and techniques at our disposal for analyzing malware.

Malware analysis is the art of dissecting malware to understand how it works, how to identify it, and how to defeat or eliminate it. And you don't need to be an uber-hacker to perform malware analysis.

With millions of malicious programs in the wild, and more encountered every day, malware analysis is critical for anyone who responds to computer security incidents. And, with a shortage of malware analysis professionals, the skilled malware analyst is in serious demand.

That said, this is not a book on how to find malware. Our focus is on how to analyze malware once it has been found. We focus on malware found on the Windows operating system—by far the most common operating system in use today—but the skills you learn will serve you well when analyzing malware on any operating system. We also focus on executables, since they are the most common and the most difficult files that you'll encounter. At the same time, we've chosen to avoid discussing malicious scripts and Java programs. Instead, we dive deep into the methods used for dissecting advanced threats, such as backdoors, covert malware, and rootkits.

Prerequisites

Regardless of your background or experience with malware analysis, you'll find something useful in this book.

Chapters 1 through 3 discuss basic malware analysis techniques that even those with no security or programming experience will be able to use to perform malware triage. Chapters 4 through 14 cover more intermediate

material that will arm you with the major tools and skills needed to analyze most malicious programs. These chapters do require some knowledge of programming. The more advanced material in Chapters 15 through 19 will be useful even for seasoned malware analysts because it covers strategies and techniques for analyzing even the most sophisticated malicious programs, such as programs utilizing anti-disassembly, anti-debugging, or packing techniques.

This book will teach you how and when to use various malware analysis techniques. Understanding when to use a particular technique can be as important as knowing the technique, because using the wrong technique in the wrong situation can be a frustrating waste of time. We don't cover every tool, because tools change all the time and it's the core skills that are important. Also, we use realistic malware samples throughout the book (which you can download from *http://www.practicalmalwareanalysis.com/* or *http://www.nostarch.com/malware.htm*) to expose you to the types of things that you'll see when analyzing real-world malware.

Practical, Hands-On Learning

Our extensive experience teaching professional reverse-engineering and malware analysis classes has taught us that students learn best when they get to practice the skills they are learning. We've found that the quality of the labs is as important as the quality of the lecture, and without a lab component, it's nearly impossible to learn how to analyze malware.

To that end, lab exercises at the end of most chapters allow you to practice the skills taught in that chapter. These labs challenge you with realistic malware designed to demonstrate the most common types of behavior that you'll encounter in real-world malware. The labs are designed to reinforce the concepts taught in the chapter without overwhelming you with unrelated information. Each lab includes one or more malicious files (which can be downloaded from *http://www.practicalmalwareanalysis.com/* or *http://www.nostarch.com/malware.htm*), some questions to guide you through the lab, short answers to the questions, and a detailed analysis of the malware.

The labs are meant to simulate realistic malware analysis scenarios. As such, they have generic filenames that provide no insight into the functionality of the malware. As with real malware, you'll start with no information, and you'll need to use the skills you've learned to gather clues and figure out what the malware does.

The amount of time required for each lab will depend on your experience. You can try to complete the lab yourself, or follow along with the detailed analysis to see how the various techniques are used in practice.

Most chapters contain three labs. The first lab is generally the easiest, and most readers should be able to complete it. The second lab is meant to be moderately difficult, and most readers will require some assistance from the solutions. The third lab is meant to be difficult, and only the most adept readers will be able to complete it without help from the solutions.

What's in the Book?

Practical Malware Analysis begins with easy methods that can be used to get information from relatively unsophisticated malicious programs, and proceeds with increasingly complicated techniques that can be used to tackle even the most sophisticated malicious programs. Here's what you'll find in each chapter:

- Chapter 0, "Malware Analysis Primer," establishes the overall process and methodology of analyzing malware.
- Chapter 1, "Basic Static Techniques," teaches ways to get information from an executable without running it.
- Chapter 2, "Malware Analysis in Virtual Machines," walks you through setting up virtual machines to use as a safe environment for running malware.
- Chapter 3, "Basic Dynamic Analysis," teaches easy-to-use but effective techniques for analyzing a malicious program by running it.
- Chapter 4, "A Crash Course in x86 Assembly," is an introduction to the x86 assembly language, which provides a foundation for using IDA Pro and performing in-depth analysis of malware.
- Chapter 5, "IDA Pro," shows you how to use IDA Pro, one of the most important malware analysis tools. We'll use IDA Pro throughout the remainder of the book.
- Chapter 6, "Recognizing C Code Constructs in Assembly," provides examples of C code in assembly and teaches you how to understand the high-level functionality of assembly code.
- Chapter 7, "Analyzing Malicious Windows Programs," covers a wide range of Windows-specific concepts that are necessary for understanding malicious Windows programs.
- Chapter 8, "Debugging," explains the basics of debugging and how to use a debugger for malware analysts.
- Chapter 9, "OllyDbg," shows you how to use OllyDbg, the most popular debugger for malware analysts.
- Chapter 10, "Kernel Debugging with WinDbg," covers how to use the WinDbg debugger to analyze kernel-mode malware and rootkits.
- Chapter 11, "Malware Behavior," describes common malware functionality and shows you how to recognize that functionality when analyzing malware.
- Chapter 12, "Covert Malware Launching," discusses how to analyze a particularly stealthy class of malicious programs that hide their execution within another process.
- Chapter 13, "Data Encoding," demonstrates how malware may encode data in order to make it harder to identify its activities in network traffic or on the victim host.

- Chapter 14, “Malware-Focused Network Signatures,” teaches you how to use malware analysis to create network signatures that outperform signatures made from captured traffic alone.
- Chapter 15, “Anti-Disassembly,” explains how some malware authors design their malware so that it is hard to disassemble, and how to recognize and defeat these techniques.
- Chapter 16, “Anti-Debugging,” describes the tricks that malware authors use to make their code difficult to debug and how to overcome those roadblocks.
- Chapter 17, “Anti-Virtual Machine Techniques,” demonstrates techniques used by malware to make it difficult to analyze in a virtual machine and how to bypass those techniques.
- Chapter 18, “Packers and Unpacking,” teaches you how malware uses packing to hide its true purpose, and then provides a step-by-step approach for unpacking packed programs.
- Chapter 19, “Shellcode Analysis,” explains what shellcode is and presents tips and tricks specific to analyzing malicious shellcode.
- Chapter 20, “C++ Analysis,” instructs you on how C++ code looks different once it is compiled and how to perform analysis on malware created using C++.
- Chapter 21, “64-Bit Malware,” discusses why malware authors may use 64-bit malware and what you need to know about the differences between x86 and x64.
- Appendix A, “Important Windows Functions,” briefly describes Windows functions commonly used in malware.
- Appendix B, “Tools for Malware Analysis,” lists useful tools for malware analysts.
- Appendix C, “Solutions to Labs,” provides the solutions for the labs included in the chapters throughout the book.

Our goal throughout this book is to arm you with the skills to analyze and defeat malware of all types. As you’ll see, we cover a lot of material and use labs to reinforce the material. By the time you’ve finished this book, you will have learned the skills you need to analyze any malware, including simple techniques for quickly analyzing ordinary malware and complex, sophisticated ones for analyzing even the most enigmatic malware.

Let’s get started.