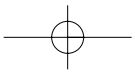

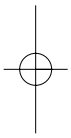

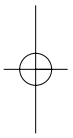

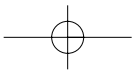

Programming Windows Security

Keith Brown

Addison-Wesley

Boston • San Francisco • New York • Toronto • Montreal
London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

The publisher offers discounts on this book when ordered in quantity for special sales. For more information, please contact:

Pearson Education Corporate Sales Division
One Lake Street
Upper Saddle River, NJ 07458
(800) 382-3419
corpsales@pearsontechgroup.com

Visit us on the Web at www.awl.com/cseng/

Library of Congress Cataloging-in-Publication Data

Brown, Keith, 1967 Mar. 16-
Programming Windows security / Keith Brown.
p. cm.
Includes bibliographical references and index.
ISBN 0-201-60442-6
1. Computer security. 2. Microsoft Windows NT. 3. Computer programming. I. Title.

QA76.9.A25 B78 2000
005.8–dc21

00-033148

ISBN 0-201-60442-6

Text printed on recycled paper.
1 2 3 4 5 6 7 8 9 10 – CRS – 04 03 02 01 00
First printing, August 2000

To my wife, Kathy; my sons, Colin, Nathan, and Aidan; and my mother, Carol. Thanks for your enduring patience and love.

Contents

Preface

As with most of my friends, I learned to program Windows by reading Charles Petzold's classic tome, *Programming Windows*. I then moved on to Jefferey Richter's seminal book for systems developers, *Advanced Windows NT*.[1] Finally, I moved into the realm of objects with Kraig Brockschmidt's *Inside OLE 2*.[2] With the release of Windows NT 4.0, I started using (and eventually teaching) COM as a way to build distributed applications. Until this point in my life, I'd been able to safely ignore security, and had long since suppressed the pangs of guilt I used to feel when passing NULL for `LPSECURITY_ATTRIBUTES`. Little did I know that my life was about to change forever.

It was a beautiful sunny day in Bellevue, Washington, when I drove up to the offices of Saros, a software development company where I was scheduled to give my first on-site presentation of Essential COM, DevelopMentor's flagship COM course that included coverage of the relatively new Windows NT 4.0 feature called DCOM. All the students in the class had packed in their own

[1] At least I think this was the title way back then…

[2] Like many early adopters, I started off with a laser-printed draft copy of Kraig's first edition, crying in my beer over what had happened to the operating system that I knew and loved.

computers (these students were worn and grizzled Windows programmers, some of whom had lost the covers to their machines long ago). What made the situation interesting was that some of the students had machines belonging to various Windows NT domains, whereas others had standalone machines not associated with any domain. One student was even running Windows 95. It was a recipe for disaster. Everything had been going smoothly, and the students (and myself) were thoroughly enjoying the class, but the radical configuration in the classroom put quite a crimp in the DCOM lab exercise that morning. Virtually all the students were getting E_ACCESSDENIED and various and sundry error codes, and of course they all looked to me to fix the problem. I failed miserably that day, and had to admit to myself (and the students) that I didn't yet have a good enough grasp of Windows security to solve their problems. I've rarely felt so small.

Shortly after this soul-wrenching experience, I dedicated myself to the pursuit of a deep and practical understanding of Windows security. I solidified my commitment by agreeing to write a new course for DevelopMentor about services and security, and proceeded to spend three months of virtually uninterrupted time studying and experimenting with the Windows security APIs. I never knew that I'd end up falling in love. Since then, I've learned loads more and answered hundreds of questions on the DCOM mailing list regarding security issues, and reached thousands of students, conference attendees, and readers of *Microsoft Systems Journal* (now *MSDN Magazine*) with the message that security is a fascinating and approachable topic.

As the culmination of my effort, this book attempts to fill the gaping hole in the Windows systems programming canon by providing a reference for programmers that covers Windows security from the basics of principals, authorities, logon sessions, and DACLs all the way through COM+ security, one of the most subtle and sensitive beasts you'll encounter as a Windows programmer.

As a side effect of my predilection for distributed programming, this book is unique in that it addresses security with the distributed systems developer in mind; in fact, the original title of the book was *Distributed Security in Windows NT*. Of course, Microsoft's decision to rename their (beta, at the time) operating system from "Windows NT 5.0" to "Windows 2000" didn't bode well for

my original title. Frankly, *Distributed Security in Windows NT/2000* sounded really goofy. Thus the new title.

In any case, if you're a programmer who, not unlike myself a few years ago, feels a hollowness in the pit of your stomach as a result of being asked to add a security-related feature or debug a security-related problem in an application, I hope you'll find that this book completes you.

Which Windows?

This book covers security programming on Windows 2000 and Windows NT 4. Therefore, to avoid crossing the reader's eyes with "Windows 2000/NT" or similar nastiness, I'll simply refer to both of these operating systems as *Windows*. If I find the need to say anything specific about a distinct operating system (including Windows 9x), I'll use the full name.

Who Should Read This Book

This book was written for professional software developers involved in systems programming on Windows. The third part of the book was written for the subset of these programmers developing distributed systems (especially those using COM).

The first part of the book (the first three chapters) intentionally has no code in it, and instead speaks to the big picture, introducing terminology and concepts that will likely be useful not just for programmers but also for technical managers and advanced Windows users. If you are a manager and want to get a better understanding of Windows security, borrow this book from one of the developers on your team and rip out the first three chapters for yourself. Sharing a common terminology will help you and your development team play better together.

What a Developer Should Already Know

I assume you have a basic understanding of Windows system programming; that is, you know the difference between a process and a thread, and you've written a DLL before and perhaps even written a service or two. I casually mention features such as *thread local storage* and assume that you know what I'm talking about. If you're unsure of your ability in this area, my favorite book on

the subject is *Advanced Windows,* by Jefferey Richter (as of this writing the fourth edition is hot off the press).[3]

In the COM chapter (Chapter 9), I assume you feel comfortable with the basics of `IUnknown` and that you know what a proxy and stub are. If you're unsure, my favorite COM book is *Essential COM* by Don Box.

Much of the later material in the COM chapter talks about COM+ features in Windows 2000, which *Essential COM* doesn't address (as of this writing, *Essential COM* is in its first edition). My favorite COM+ book in print as I write this is *Programming Distributed Applications with COM and Visual Basic 6.0* by Ted Pattison.[4] Tim Ewald's book *Transactional COM+: Designing Scalable Applications* will likely be a must-read as well, although it's not yet gone to press as I write this.

Although this book often shows full declarations of Windows API functions, I won't always bother to tell you the details of what each and every parameter means if it's not relevant to the topic at hand. I hope you find that this book works well as a *complement* to the Windows API documentation, as opposed to a *replacement* for it.

How to Approach This Book

I know that most developers don't relish thinking about security issues, which is often why security ends up getting retrofitted into existing products (or left until the last minute in new products). Most of my students are really surprised to learn that security is actually quite an interesting topic, and they really enjoy sitting through DevelopMentor's security class. However, I'm aware that this is a self-selecting group; these folks have *chosen* to take the class, often because there is a distinct set of problems that they need to be able to solve, whether they like it or not. Whatever your predisposition is toward security, I designed this book to be readable front to back, but also to be readable in chunks.

Many people will buy this book because it contains (at least as of this writing) an exhaustive coverage of COM security, and will want to dive right in to the chapter on COM. However, you can't possibly understand COM security

[3] Although the title has changed to *Programming Applications for Windows,* 4th ed.

[4] Ted is working on a new edition as this goes to press, so keep your eyes peeled for it.

without having a basic understanding of the fundamentals, and no matter how much I urge folks to read Chapter 4, there will be a large group of people who don't have the time for this. If it's you I'm describing, do spend the time to read the first three (very short) chapters of this book before you start diving into the nitty-gritty details of COM security. These chapters will help you develop a more intuitive feel for how Windows security works and why it works the way it does.

This Is *Not* a Cookbook

In the vein of my last book project, *Effective COM* (coauthored with Don Box, Tim Ewald, and Chris Sells), I've purposely avoided making this a cookbook that provides lots of code for you to cut and paste to solve a particular set of problems that you may or may not be faced with. Instead, this book is about helping you understand how things work. I'd love to see a Windows security cookbook written. I find that cookbooks increase my long-term productivity once I have a basic understanding of the topic at hand.

The code snippets in this book should all compile correctly. However, although some of them might be appropriate for cutting and pasting into your projects, be aware that a number of them exist solely to foster insight and understanding and will necessarily be a bit more abstract than what you'd expect to find in a cookbook.

The Bad Guys

Often I'll refer to the "bad guys" when I want to indicate someone who may be trying to break into your system either to do malicious damage, or just for fun. The bad guys are the folks that we want to keep out, and the good guys are folks that we want to let in. I actually borrowed the terms "good guys" and "bad guys" from one of my favorite security books, *Network Security: Private Communication in a Public World* (Kaufman, Perlman, and Speciner 1995).

Code Sample Conventions

All the code samples in this book were built using Visual C++ 6 with the Platform SDK for Windows 2000. Each snippet was compiled, and all functional samples were tested as well. This occurred *before* they were copied into the manuscript, so any syntax errors you find are likely printing errors.

Download the code snippets from http://www.develop.com/books/pws to get the real thing.

I built all the code samples with the UNICODE macro defined; I didn't want to clutter the code with __TEXT macros when this book is all about programming Windows 2000 and Windows NT 4, where Unicode is the norm.

I use a consistent naming scheme for any functions that are my own, so that you can distinguish them from system calls. My functions (and constants) all start with a lowercase letter prefixed with an underscore:

```
_thisIsACallToMyFunction();
ThisIsACallToASystemFunction();
```

Let me warn you that the snippets I provide sometimes ignore error checking for brevity, except in certain cases where I have some special insight to offer or when I'm providing functions that are generally useful to be worthy of direct cut-and-paste. There are many ways of performing error checks (many wars have been fought over the correct way to do this), but virtually all error checking mechanisms obscure the system calls you're making to some degree, and it's these calls that I want to focus on in the code snippets.

Finally, I'm a real stickler when it comes to writing const-correct production C++ code, but I found that this doesn't work well in the limited space a book affords. (Much of the Windows security API is notoriously const-incorrect, and having you wade through oodles of const_cast operators wouldn't serve any good purpose.)

Yes, There Is No CD-ROM

I've been told that we live in the "information age," and I personally think it's silly to ship a CD with stale content when I am perfectly willing to provide up-to-date content via the Web. So please visit http://www.develop.com/books/pws to download real examples that compile and build (this includes all the code snippets from this book, plus lots of other goodies that I upload from time to time).

Errata

I've gone out of my way to research all the topics in this book, but as with any endeavor of this magnitude, there's bound to be a few rough edges. Please send

any errata to me via my Web site (http://www.develop.com/books/pws). I'll publish all confirmed bugs online, and credit the first person to report the problem. Please check my Web site from time to time to keep abreast of any problems that may have been reported.

What to Expect

Part I: Model

These first chapters were written to give you a roadmap of the Windows security architecture. These chapters are designed to be as concise as possible so that a dedicated reader can consume them comfortably in one or two sittings. My goal is to introduce some basic terminology with an emphasis on how all the pieces fit together, without drilling down into the details. An effective way to use this part of the book would be to read through it once before diving into the other chapters in the book, and then revisit these chapters whenever you need to step back and see the big picture. There is no code in these chapters, so this is a great section to tear out and send to your manager to help bridge the communication gap that often develops on a project.

Chapter 1: The Players

This chapter focuses on the actors in a secure system. It introduces principals and authorities, authentication, domains, and the Local Security Authority (LSA). The chapter emphasizes that security eventually boils down to trust, and provides several examples.

Chapter 2: The Environment

This chapter focuses on the environment in which your programs run. It introduces logon sessions, tokens, window stations, and profiles.

Chapter 3: Enforcement

This chapter focuses on authorization and access control. It introduces groups, aliases, roles, privileges, security descriptors, and DACLs and SACLs, as well as some access control strategies and guidelines for picking an appropriate strategy for your application. The chapter ends with a discussion of the session-oriented nature of Windows security.

Part II: Mechanics

These next three chapters drill down into the details of each of the concepts introduced in Part I. Except where noted, you can read these in pretty much any order you like.

Chapter 4: Logon Sessions

This chapter delves into the details of logon sessions and tokens. Systems developers will feel much more comfortable designing and implementing applications with a good grasp of logon sessions. This chapter discusses the System logon session, as well as interactive and network logon sessions and how to call `LogonUser` to establish new logon sessions. It also shows how to make use of privileges at runtime and restrict privileges with job objects.

Chapter 5: Window Stations and Profiles

Many Windows developers have never even heard of a window station, but these seemingly obscure entities will eventually haunt you if you don't come to grips with them. This chapter includes a discussion of window stations and desktops, as well as a discussion of user profiles and how to manage them. To get the most out of this chapter, read the previous chapter on logon sessions first.

Chapter 6: Access Control and Accountability

This chapter shows how to create and manage security descriptors, including access control list (ACL) programming and auditing. ACLs in Windows 2000 change quite dramatically from those in earlier versions of the operating system, and these changes are covered in detail. The chapter also includes a discussion of how to manage and use private security descriptors for securing application-defined objects, including dealing with object hierarchies and ACL inheritance.

Part III: Distribution

Parts I and II deal with basic Windows security programming. Part III builds on this foundation by showing how distribution factors into the security model. Many companies are developing Windows-based distributed systems these days, and most of them rely on COM and HTTP as integral enabling technolo-

gies. This book therefore culminates in a discussion of COM and IIS security. Except where noted, you can read these chapters in any order you like.

Chapter 7: Network Authentication

The problem of proving one's identity to another across a public wire was the primary question that originally captivated me and initiated my love affair with security. It's a fascinating problem with many solutions, and this chapter provides an introduction to the core network authentication protocols used in Windows NT and Windows 2000, namely, NTLM and Kerberos. After describing and contrasting the two protocols, the chapter concludes by introducing the Security Support Provider Interface (SSPI), which abstracts the differences between various authentication protocols.

Chapter 8: The File Server

Using the Windows file system across the network is a very common practice, and this chapter is dedicated to exploring the sort of security programming problems you're likely to encounter in these scenarios. This chapter is all about understanding SMB (Server Message Block) security and how to bend it to your will. Because named pipes are built on top of the file server infrastructure, I've also included them in this discussion.

Chapter 9: COM(+)

This chapter draws on the basics introduced in earlier chapters to provide the foundation for a solid understanding of COM(+) security, one of the most misunderstood and oft-cursed features in Windows. I address COM+ security features and provide notes on differences between COM+, MTS, and base COM. You'll get the most out of this chapter if you've already read the first and second parts of this book (Chapter 4 is the most important chapter from the second part). I'd also recommend reading Chapter 7 before tackling this chapter.

Chapter 10: IIS

DCOM isn't a popular protocol for use over the Internet. In fact, just getting it to cross firewalls and network address translation layers is quite a feat of engineering. The Internet is about simplicity, and HTTP and SSL are the protocols

of choice for reaching the broadest audience. Often a distributed system is built using DCOM in the middle tier, with HTTP used as a gateway to the client tier. This chapter first covers the basics of SSL and certificate-based authentication, and then turns and focuses on issues you need to be aware of when building Web applications with IIS, especially when coupling them with a middle tier of COM+ components. The latter parts of this chapter will make more sense if you've read Chapter 9.

Appendix: Some Parting Words

I've put together some tips for writing setup programs (how to install user and group accounts, configure privileges, and configure secrets such as the COM RunAs password). I've included a list of well-known SIDs that you can form programmatically, and a simple class for making this easy to do. I've also included a discussion of the three different group scopes in Windows 2000 (universal, global, and domain local). Finally, I've included a list of all the defined privileges in Windows along with as much insight as I could muster into how they really work (the documentation often is too vague to be of use).

Glossary

Throughout the text, new terms are called out in bold as they are introduced, and are summarized in the glossary. I hope you find this section helpful.

Bibliography

Any book or magazine articles I reference can be found in the bibliography.

What Not to Expect

Active Directory

I'll spend just enough time talking about the Active Directory Services Interface (ADSI) to get you started installing user and group accounts. An entire book could (and should) be written on programming the Windows directory. This is not that book.

Public Key Infrastructure

Although I discuss SSL authentication and the basics of how certificates work, any detailed coverage of what it takes to build a public key infrastructure is beyond the scope of this book. I provide several references in the bibliography for interested readers.

Acknowledgments

First of all, I want to acknowledge the tremendous sacrifice my wife and kids made while I struggled to get this book to press. I've hardly seen them at all for the last four months of this project. Thanks Kathy, Colin, Nathan, and Aidan. I've missed you so incredibly much.

I'd like to thank Don Box and Mike Abercrombie of DevelopMentor for providing an unparalleled environment for research and development, and for feeding my family while I hunkered down to finish the book. I've had a ton of fun working with you guys, and I look forward to many more years of collaboration.

Thanks to Bruce Schneier for writing an incredibly readable book on cryptography that captivated me. Reading *Applied Cryptography* was a turning point in my life, because I discovered what a fascinating game security really was.

Thanks to all the students in my security classes over the past couple of years who have listened to an evolving story and have provided their own unique input. This story wouldn't be the same without you.

Thanks to the reviewers who gave me feedback on this project: Saji Abraham, Richard Ward, Michael Howard, Bob Beauchemin, Ian Griffiths, George Reilly, Michael Nelson, Steve Rodgers, Thomas Deml, Henk de Koning, and Jefferey Richter.

Thanks to the staff at Addison Wesley: Kristin Erickson, for being an advocate and friend, Jacquelyn Doucette for pushing production through in record time, and J. Carter Shanklin for talking me into this thing in the first place.

Thanks to my copyeditor Cindy Kogut, who continues to amaze me with her ability to cover my prose with oodles of red ink. Cindy also retrofitted a healthy dose of consistency into a book whose conception spanned two years of my life.

And finally, thanks to Alice and Bob for just being you.