book reviews

ELIZABETH ZWICKY, ERIC SORENSON, AND SAM STOVER

THE .NET DEVELOPER'S GUIDE TO WINDOWS SECURITY

Keith Brown

Addison-Wesley, 2004, 392 pages.
ISBN 0-321-22835-9.

OK, so this is not an obvious review choice for me. I'm not a developer, I don't do .NET, and I don't do much Windows—which leaves "Security" as the only relevant word in the title. To be honest, I didn't request the book, for no very obvious reason the publisher sent it along with books I did request, and I picked it up mostly because I thought "Hey, it's short and irrelevant; I can glance at it, determine that I don't care, and move on rapidly."

So I picked it up and read a random page, which to my surprise was lucid and interesting. So I read the whole thing; and then I read bits of it aloud to my colleagues, who said things like, "Hey! Can I borrow that?" even though they aren't .NET developers either.

If you happen to know any .NET developers, you should run out, buy this book, and force them to read it. Then force them to reread the bits on running as a normal user and developing software that system administrators don't hate. If you are a system administrator, you should definitely read those bits yourself, just so you can have the warm fuzzy feeling that out there somewhere is a developer who understands.

And if you need to know how security-relevant parts of Windows actually work, buy this book even if you have no interest at all in .NET, because it's full of clear explanations and practical tips. I was awe-struck by the explanation of why group nesting works the way it does. I mean, I suppose I had always assumed there was some reason, but nobody had ever mentioned one, let alone diagrammed it out. And now I have been enlightened.

LEARNING WINDOWS SERVER 2003

Jonathan Hassell

O'Reilly, 2006. 723 pages.
ISBN 0-596-10123-6.

THE ULTIMATE WINDOWS SERVER 2003 SYSTEM ADMINISTRATOR'S GUIDE

Robert Williams and Mark Walla

Addison-Wesley, 2003. 956 pages.
ISBN 0-201-79106-4.

As I said above, I don't do Windows much. In fact, the Windows 2003 machines I administer are entirely virtual (users and all), and if something goes wrong, I cheerily destroy them and start another nice clean one that a real Windows administrator built an image for. Thus, I'm reviewing these books from the point of view of a UNIX administrator with a need to understand things periodically, not of somebody who lives and breathes Windows system administration.

However, I have had reason to consult these books recently, and there's a clear pattern. *Learning Windows Server 2003* has the information I need, without a lot of other stuff, and with handy information about command-line tools. The *Ultimate Windows Server 2003 System Administrator's Guide* has more deployment information, but it's less clearly written and often obfuscates rather than clarifying important details. For instance, it tells you that users can read files even when they don't have permission to read the folder the files are in, but it doesn't mention that this is in fact a user right that can be revoked. It has lots of illustrations of the dialogs used to set up groups, and of particular group configurations (many of which it tells you not to use), but it doesn't have a table that says what kind of users and groups can be members of each kind of group. *Learning Windows Server 2003* has such a table. The *.NET* book also has such a table, plus an explanation of why the groups work the way they do and what the pros and cons of each kind of group are.

Here's how each book introduces groups:

The .NET Developers Guide to Windows Security: "Most developers have a basic idea of what a security group in Windows is all about. It's a way to simplify administration by grouping users together. In a large system, a clearly defined group can allow an administrator to assign permissions for hundreds of files, registry keys, directory objects, and so on, without having to think about each individual user that will be in the group to which he or she is granting permission."

Learning Windows Server 2003: "The point of groups is to make assigning attributes to larger sets of users easier on administrators. Picture a directory with 2,500 users. You create a new file share and need to give certain employees permission to that file share (e.g., all accounting users)."

The Ultimate Windows Server 2003 System Administrator's Guide: "A group is a collection of

users, computers, and other entities. It can be a Windows Server 2003 built-in group or one created by the system administrator to conform to required attributes. Windows Server uses standard groups to reflect common attributes and tasks."

As you can see, the books have very different styles. I recommend *Learning Windows Server 2003* as a general reference, and *The .NET Developers Guide to Windows Security* for understanding underpinnings.

THE BEST SOFTWARE WRITING I

Joel Spolsky, Editor

Apress, 2005. 305 pages.
ISBN 1-59059-500-9.

This is a collection of essays, some of them brilliant, some of them thought-provoking, some of them laugh-out-loud funny, and (from my point of view) one clunker (no, I won't tell you which one). Most of them are available on the Web already, although if you've found them all you read way too many blogs.

The canonical reader here is a senior programmer in a startup, but anybody who likes to think about programming, programming languages, or the computer industry will find something to chew on and something to laugh at. Or maybe weep at—I'm honestly not sure whether the appropriate response to "Powerpoint Remix" is to laugh or to cry.

If you're looking for a great airplane book, this is quick but meaningful stuff. I enjoyed it vastly.

EXTRUSION DETECTION: SECURITY MONITORING FOR INTERNAL INTRUSIONS

Richard Bejtlich

Addison-Wesley Professional, 2005.
ISBN 0-321-35996-2.

REVIEWED BY ERIC SORENSON

Extrusion Detection is, in sum, a refocusing of the methodology Bejtlich detailed in *The Tao of Network Security Monitoring*. Here he shifts the emphasis from threats that generate traffic out on the Internet to ones that are already inside an enterprise's perimeter and must be detected by inspecting outbound traffic. Since *Extrusion Detection* ought to be useful by itself without requiring *The Tao of NSM*, Bejtlich necessarily repeats some material but fortunately provides enough new information to make for some worthwhile reading, even for experienced NSM practitioners.

He begins by introducing general network security and intrusion detection principles along with the NSM credo: "Prevention eventually fails." The goal of network security monitoring is to provide a framework for answering the dreaded question, "We've been compromised—What now?" As such, the bulk of Part I of *Extrusion Detection* describes specific measures an administrator can put in place to produce good answers: blocking and proxying outbound traffic, placing packet capture sensors at choke-points, and using router features such as null routes and reverse path forwarding. With these in place, the administrator will run a "defensible network," that is, one that gives a reasonable chance of dealing with an intruder.

In Parts II and III, Bejtlich walks through what "dealing with an intruder" might entail, first as a procedural framework (Part II) and then with a specific extrusion in the form of an unauthorized botnet (Part III). Unfortunately, here the book's greatest strength is offset by its biggest weakness. The sections that detail specific steps and principles to observe during incident response and when gathering network forensics data could be powerfully useful references for one of the primary target audiences of the book—savvy enterprise network administrators who have not had much opportunity to do formal incident response. But instead of presenting the information in capsule, checklist form and then providing detail, there is lengthy—in some cases multiple-page—output from commands and sample data between steps, which makes it difficult to get a good overview of the entire process. In an emergency, the book would be frustrating if not outright dangerous to thumb through as you're responding. (The author's three-page description of his difficulty reading a mangled pcap file in the Network Forensics chapter is a prime example of this problem.)

However, this should not detract from the overall thumbs-up I give this book; a prepared admin team will have customized the book's suggested incident response procedure for their environment and summarized it (on paper!) before anything actionable happens—right? There is plenty of useful information in *Extrusion Detection*: the comparisons of SPAN monitor ports versus network taps, the routing and filtering tricks, and the walkthrough that shows the discovery of a botnet are worth the price of admission. The exploits that lead to a high-profile Web server's defacement get most of the press, but Richard Bejtlich's *Extrusion Detection* describes a methodology for dealing with threats that are potentially far more damaging, because they move from the inside out.

BUFFER OVERFLOW ATTACKS

James C. Foster, Vitaly Osipov, Nish Bhalla, and Niels Heinen

Syngress, 2005. ISBN 1-932266-67-4.

REVIEWED BY SAM STOVER

When I first started amassing my "Foster Library," I was pretty excited. I couldn't wait to find the time to really sink my teeth into the guts of buffer overflows and exploit code. I still have that desire to learn, but as I start plowing through the books, I'm becoming more and more disappointed with the library. The book entitled *Buffer Overflow Attacks* (BOA) was written in 2004 and provided the foundation for a more recent book I've reviewed previously called *Writing Exploits and Security Tools* (WSTaE). Well, it's not just the foundation, but the house, garage, yard, trees in the yard, birds in the trees, etc., etc.

I think saying that the overlap between these two books borders on the criminal is not an overstatement. I think saying that WSTaE is an updated version of BOA is misleading. True, some of the grammar and wording has been changed, such as the shift from "commonest" to "the most common." Some of the chapter titles have changed: "Buffer Overflows: The Essentials" has become "Writing Exploits and Security Tools," and "Stack Overflows" is now "Exploits: Stack."

Other than that, not much has changed. WSTaE actually does have more content with chapters devoted to Metasploit, Nessus, and Ethereal, but the core of the book is so "cut and paste" that over half of WSTaE is completely redundant. It would have made more sense to omit the overlap and release an update to BOA—but at a drastically reduced price.

There are a couple of gems in BOA that aren't in WSTaE, but I'm not convinced that they are worth the $35. There are three sections dedicated to case studies, which walk through 11 exploits as well as providing an introduction to Inline Egg (which is discussed in much greater detail in WSTaE).

This commonality puts me in a rather awkward position—there isn't much to review that I haven't already discussed in a previous review. The crux of the matter, then, is to help you the reader and potential purchaser to make a decision as to which book fits your needs. If you've already purchased BOA, then you're bound to be disappointed when you have to throw down another $50 for WSTaE, which does have a lot more information, notably the chapters on Metasploit. However, if you already own the *Penetration Tester's Open Source Toolkit* (Syngress, 2005), then you have two of the three chapters on Metasploit. Sheesh. If you already own WSTaE, you'll definitely want to think carefully before ordering BOA—at the very least cruise over to your local bookstore and see if the case studies are really worth your hard-earned dollars. If you don't own any of these books yet, and are looking for a first purchase, don't waste your time with BOA, go directly to WSTaE.

In conclusion, I have to say that *Buffer Overflow Attack*s was a pretty big disappointment. The incestuous relationship between Mr. Foster's books leaves me with a sour taste in my mouth. Writing exploits is a hot item right now, and ripping content from one book seems a direct attempt to exploit the unwary novice (pun intended).