book reviews

ELIZABETH ZWICKY

zwicky@greatcircle.com

ADAM TUROFF

ziggy@panix.com

SILENCE ON THE WIRE: A FIELD GUIDE TO PASSIVE RECONNAISSANCE AND INDIRECT ATTACKS

Michal Zalewski

No Starch Press, 2005, 1-59327-046-1, 281 pp.

Reviewed by Elizabeth Zwicky

Zalewski's book is an interesting tour through some security-related stuff. It is not, as the subtitle implies, an overview of passive attacks. It's more of a country ramble than the field guide you might use on such a walk. It wanders through security, and computing as a whole, with some novel information, some basic information you may not have come across before, and some information you already knew and didn't much care about. I, for instance, found the description of how modems actually work irrelevant but amusing, but yawned my way through the state tables that accompany the recapitulation of the entire history of computing, starting with logic itself and ending with modern processor design. Then again, I was fascinated by the section on random number generators; it's a frequently discussed topic in security, but it's still rare to see a lucid discussion of how they work and don't work, and why it matters. Plus, the pictures are very convincing.

If you are looking for practical information of immediate use about passive and indirect attacks, this is not the book for you; then again, it makes no claim to be. If you are a "hacker" type in the old sense of the word, fond of taking things apart to see how they work, and you have any interest in security, you will probably find significant portions of this book intriguing. Try not to be turned off by the initial chapters, which unfortunately are the weakest.

OPTIMIZING LINUX PERFORMANCE: A HANDS-ON GUIDE TO LINUX PERFORMANCE TOOLS

Philip G. Ezolt

Prentice Hall, 2005, 0-13-148682-9, 353 pp.

PERFORMANCE TUNING FOR LINUX SERVERS

Sandra K. Johnson, Gerrit Huizenga, and Badari Pulavarty, eds.

IBM Press, Pearson plc, 2005, 0-13-144753-X, 547 pp.

Reviewed by Elizabeth Zwicky

This batch of books brought me two on Linux performance optimization, an under-served topic, and a difficult one to write about. Performance tuning is difficult, and doing it well involves three things:

1. Excellent detective skills.

2. An intimate understanding of precisely how the system you are tuning works.

3. Familiarity with applicable tools.

1 is extraordinarily difficult to teach at all, let alone in a book. 2 is an immense amount to cover, differs importantly between sub-releases of a single product, and leads into territory best covered by "Internals of" books. That leaves 3, which isn't all that interesting without 1 and 2.

Optimizing Linux Performance does a nice job of negotiating these difficult waters. It's not going to turn anybody into a master performance tuner (no book is), but it should give somebody with normal system administration skills sufficient background in the tools and techniques to start solving problems. Its descriptions of how things work are brief, but deep enough that you can understand what the tools are doing and why you care, and it covers a wide range of tools, with information about how to use their output. Most valuably, it talks about, and demonstrates, the process of performance tuning—how you actually put the tools together to get a desired result.

Performance Tuning for Linux Servers does not succeed so well. It attempts a vast and ambitious task: to cover everything you need to know about Linux internals, down to very precise levels of detail, plus performance characteristics of all the common kinds of server workloads, and case studies, with a tutorial on programming for performance thrown in. I'm prejudiced to start with against books with large numbers of authors, and this one, with 23 authors and three editors, has only strengthened my prejudices. The chapters are of uneven quality and often overlap or even contradict each other. Many of the chapters appear to have originally been written for other purposes and fail to integrate well with the rest of the book.

This book contains an enormous amount of information, but it's not in a form that is particularly usable. I like the chapter on file servers (it's entitled "File and Print Servers," but it doesn't really say much about print servers), and the final "case study" chapter is a genuine case study that ties things together and shows a realistic troubleshooting process. The detailed sections on internals may also be

of interest if you are reasonably familiar with kernel internals on other systems and want a technical overview of Linux internals. Although the information in them is of use in performance tuning, they do not in general tell you why or how.

If you are experienced in performance tuning and want to know about Linux specifics, parts of this book may be of use to you. If you are not, the book is unlikely to be useful, and some chapters contain information that is misleading or downright incorrect: "A module is a kernel feature that provides the benefits of a microkernel without a penalty," for instance. Skip it, and get *Optimizing Linux Performance*.

PETER VAN DERLINDEN'S GUIDE TO LINUX

Peter van derLinden

Pearson Education, 2005, 0-13-187284-2, 624 pp.

LEARNING UNIX FOR MAC OS X TIGER

Dave Taylor

O'Reilly, 2005, 0-596-00915-1, 260 pp.

MAC OS X TIGER FOR UNIX GEEKS

Brian Jepson

O'Reilly, 2005, 0-596-00912-7, 395 pp.

Reviewed by Elizabeth Zwicky

Here's a batch of books trying to introduce particular UNIX variants to various audiences. *Peter van derLinden's Guide to Linux* is intended for people converting from Windows; *Learning UNIX for Mac OS X Tiger* is for experienced Macintosh users learning to use the UNIX command line; and the audience for *Mac OS X Tiger for UNIX Geeks* is experienced UNIX developers converting to the Macintosh.

Peter van derLinden's Guide to Linux might possibly have allowed my father to switch to Linux (after years of loathing Windows and using it anyway, my father has converted to a Macintosh, and getting him to use anything else is now a lost cause). But my father is pretty technological to start with. The hypothetical mother in sentences like "Even my mother can use this computer" is not going to use this book. It would be great for your reasonably technology-savvy relatives, the people who are actually running the latest version of Windows, and not because it was installed on the computer when they bought it, either, or the ones who successfully put together their own machine. It's not going to work for the people who think you're a crazy UNIX bigot who hates Microsoft for some stupid political reason of your own. Peter van derLinden tries to be balanced and fair, but he's got a very strong UNIX and open source accent, which I find charming but which will set off alarm bells in people who trust Microsoft and distrust the open source community. I'm not sure it's possible to sell Linux to those people, but I'm sure this book won't do it, although it does try.

The book covers Linux installation, using Linspire, and most day-to-day operations. It's strongly oriented to using free (both open source and non-commercial) tools, so it covers Lphoto and OpenOffice, but doesn't talk about emulation or about commercial software for Linux. It does have thorough coverage of dual boot options.

Learning UNIX for Mac OS X Tiger is what you offer to your geeky relatives with Macintosh systems to get them to use the command line. It is a solid introduction to UNIX basics with a Macintosh accent, including coverage of Macintosh-specific features and difficulties and comparisons with the GUI tools. It moves at a pretty fast clip, as you need to if you're going to get the basics of any UNIX into a mere 245 content pages; so once again, if you have the hypothetical mother people talk about, it's not going to make her happy about firing up a terminal. Your equally hypothetical technology-oriented niece, however, will be writing shell scripts in no time. I found one unfortunate and puzzling error (the statement in Chapter 2 that the shell waits for background processes to finish before returning its prompt—it's not even clear to me what this was intended to say, but it's definitely wrong), but single errors like that can occur in even the best of books.

Max OS X Tiger for UNIX Geeks is a nice introduction to software development on OS X Tiger. It starts with a general introduction to the features of MacOS X that are different from other operating systems, and then dives into what you need to know to build software. I found the general introduction enlightening and useful, although from a system administration point of view it has some shortcomings—I really wanted to know how inetd.conf plays with the other ways of starting programs on demand, for instance. These are minor issues, however, and the book is intended for developers, who can just use an appropriate Tiger method to start their programs. Even for a non-developer, it's a great reference to porting and packaging software for Tiger. I recommend it.

PERL BEST PRACTICES

Damian Conway

O'Reilly, 2005, 0-596-00173-8, 544 pp.

Reviewed by Adam Turoff

Conway's collection of 256 tips for writing better Perl programs is a breath of fresh air, highlighting practices that work well and idioms to be avoided at all costs. All of his advice is prefaced with an implicit zeroth-law of programming: *Always code as if the person who ends up maintaining your code will be a violent psychopath who knows where you live.*

Some advice, such as the chapters on formatting, naming, and control structures, may seem nitpicky and arbitrary. Taken together, they constitute a set of conventions that allow you to do more with less effort. Later chapters focus on more advanced topics where the practices described represent the difference between keeping your sanity and spending extreme amounts of time mired in needless debugging.

I recommend *Perl Best Practices* to anyone who deals with Perl even on a casual basis. This is the first book that simply and concisely captures the lore of what makes a good Perl program good, and how to avoid features that slowly lead to bit-rot. This book is already on my list of books to reread annually.

ADVANCED PERL PROGRAMMING, 2ND ED.

Simon Cozens

O'Reilly, 2005, 0-596-00456-7, 304 pp.

Reviewed by Adam Turoff

The first edition of *Advanced Perl Programming* covered an esoteric grab bag of topics, some useful, some arcane, and some impractical. These represented the vanguard of what "advanced Perl" meant in 1997. Since then, topics such as object-oriented Perl, references, closures, modules, and eval are now widely understood and no longer "advanced" topics.

If you wrote off this title based on experience with the first edition, look again. The second edition of *Advanced Perl Programming* is a complete rewrite, and attempts to address the shortcomings of the first edition. Topics considered advanced eight years ago are now taken as given. The focus is now on advanced *uses* of Perl, not advanced (and oblique) *features* of Perl. The book opens with a discussion of some deep, dark magic-playing with the symbol table and incorporation of useful features first found in other programming languages. From there, the book switches to a Cook's Tour of CPAN, where Cozens compares and contrasts alternative CPAN modules for doing similar tasks. The discussions are useful and informative, but never very deep.

While this book may continue to be mistitled, it is certainly a welcome addition to my Perl bookshelf. I recommend this book to programmers who are starting to delve into advanced topics such as parsing, templating, testing, event-centric programming, Unicode, or database modeling.