book reviews

ELIZABETH ZWICKY

zwicky@greatcircle.com

with Sam Stover and Rik Farrow

THE TCP/IP GUIDE: A COMPREHENSIVE, ILLUSTRATED INTERNET PROTOCOLS REFERENCE

Charles M. Kozierok

No Starch Press, 2005. 1,539 pages. ISBN 1-59327-047-X

When I started to review books, my husband found that several of his cherished illusions about book reviewing were shattered. First, having publishers send us free books was not as exciting as he had hoped. Second, he had believed that reviewers always lovingly read every page of every book. As an author, I've never believed that, and have in fact cherished the theory that any reviewer who dislikes my book just didn't pay sufficient attention. Different reviewers have different standards; I feel that it's necessary to read every page, except in truly extreme circumstances, but I'm willing to gloss rapidly over some of them.

At 1,539 pages, meeting that standard for the *TCP/IP Guide* has taken me quite a long time. And it's probably not representative of what other readers will do; the book is not intended to be read end-to-end like a novel. But I found that I actually got fonder of the book as I kept going. My early experiences were marred by an indexing issue (I tried to look up the port number for DHCP, which isn't indexed, although it turns out the information is there) and a fundamental disagreement with the author about what constitutes a protocol (I'm sorry, but I know of no coherent definition of the term which allows network address translation to be considered a network protocol). But as I went along I found that while I have issues with the book, it's actually informative and easy to read, even when discussing rather nasty protocols, and when it covers something, it generally covers it quite completely.

The book is something of a strange beast. I would have made some different choices about what to include and what to leave out; for instance, I've seen some pretty odd things on networks—including noncontiguous netmasks, which Kozierok asserts were never used—and I've never seen the ICMP traceroute message type in use. He does point out that it never made it out of experimental status, but only after two pages of discussion. More important, the book's only gesture towards non-UNIX systems is to discuss implementations of UNIX-based TCP/IP protocols. There's minimal coverage for Microsoft extensions and oddities, and no coverage at all of Microsoft file sharing or naming. But it's not just Microsoft that gets shorted; there's no AFS, Kerberos, or LDAP, and RPC is mentioned very briefly in passing during the discussion of NFS.

In the protocols that it does discuss (and there are lots of them), I would have made some different choices about the information to put in, preferring less history and more security, for instance.

So why is it twice as long as volumes 1 and 3 of *TCP/IP Illustrated*, which cover basically the same protocols? Well, it covers IPv6 quite thoroughly, it assumes less expertise on the reader's part, and it covers some topics (like much of the theory behind routing) that *TCP/IP Illustrated* leaves for more specialized books. If you don't have a TCP/IP background, and you're looking for understandable, implementation-neutral descriptions of protocols, it's a good choice for a reference work. Despite my initial misgivings (and my continued pedantic snarling), I'm going to give this one a place on my bookshelf.

OPEN SOURCE FOR THE ENTERPRISE: MANAGING RISKS, REAPING REWARDS

Dan Woods and Gautam Guliani

O'Reilly, 2005. 217 pages. ISBN 0-596-10119-8

You love open source, but you're not sure how to get it into your IT shop; it's not that everybody is committed to what you've got now, but they're nervous about something that seems to involve too many hippies and fanatics. Or, for that matter, you don't love open source, you're a traditional IT manager trying to figure out what to do about open source for one reason or another (not enough money to buy commercial, you're surrounded by hippies and fanatics, your vendor just snapped your last nerve). This sensible book is a good place to start. It's very much in favor of open source software, while maintaining a good grasp of the pitfalls involved, and it speaks in language that nicely bridges the worlds of open source and IT manager.

It is a small book; it left me feeling hungry for more. But it does a nice job of filling its niche. If you need to figure out how much you can reasonably do with open source, or how to convince people to do that, and you're dealing with people who think in traditional IT terms, this book will point you in the right direction and reassure you that it is possible and reasonable.

DESIGNING INTERFACES

Jennifer Tidwell

O'Reilly, 2005. 331 pages.
ISBN 0-596-00803-1

I am by no means an interface designer. On the other hand, I've ended up designing my fair share of interfaces, either because I was the only option or because everybody else involved in the project was even less able. This has left me with the unsurprising insights that interface design matters, it's a lot of work, and that people who do it seriously are better at it than I am. So I was enthusiastic about the idea of a book that would either improve my ability or at least allow me to take a reasonably interested programmer on a team and get them to my level of semi-competence.

Happily, I believe this book meets both goals. It's a book of user interface patterns meant for people who are just starting to think about the design of user interfaces. If you're a serious human-computer interaction person, it's going to be way too basic for you. If you were hoping somebody would just tell you what to do and get it over with, it's going to be too fuzzy for you. But if you're willing to do your own thinking and need somewhere to start, this book should give you the tools to work with.

INTERNET FORENSICS: USING DIGITAL EVIDENCE TO SOLVE COMPUTER CRIME

Robert Jones

O'Reilly, 2006. 216 pages.
ISBN 0-596-10006-X

Internet Forensics is somewhat misleadingly titled. If you're hoping to find out what professionals do when they track down serious crimes, or you're already familiar with computer security, you're likely to find it disappointing. It's a sensible, interesting book on amateur Internet forensics, the sort of thing you might do at home to track down people who are really annoying you. I enjoyed it, although as somebody who already has a security background, I didn't find anything particularly novel in it.

I recommend this book if you don't know a lot about security and want to do something about nasty mail and Web pages. It's also a great lesson in a bunch of basic parts of the Internet; if you want a really motivating way to learn how IP and DNS and HTTP work, it's a lot more fun than reading abstract descriptions, and it will give you a good reason to play around with things until they make sense to you.

SARBANES-OXLEY IT COMPLIANCE USING COBIT AND OPEN SOURCE TOOLS

Christian B. Lahti and Roderick Peterson

Syngress, 2005. 333 pages.
ISBN 1-59749-036-9

This book takes on challenging territory. "Sarbanes-Oxley" and "COBIT" (Control Objectives for Information and related Technology) are the sort of words that inspire simultaneous terror and boredom. Anybody involved in trying to comply with Sarbanes-Oxley is probably going to turn to COBIT as a way of getting a handle on things, but they don't map perfectly, so it's a confusing mess where the only possible downside for getting it wrong is huge fines and jail time. Just the kind of situation in which you'd like a good book to come along and hold your hand, and all the better if it includes the tools you need.

Unfortunately, this book doesn't do a particularly good job of hand-holding. It gives a nice introduction to the issues involved in Sarbanes-Oxley and COBIT, and how the two relate (although I could have done without the intro that portrayed the reader as too technology-obsessed to even pay minimal attention in important meetings; thanks, but I get enough insulting stereotypes from people who're not trying to sell books to me). After that, things go downhill. There are a lot of statements and not a lot of the sort of scaffolding you'll need to make your own decisions about your own site.

Open Source for the Enterprise (see above) does a much better job of discussing the issues and advantages of open source to meet whatever needs you have. Coverage of open source consists of a brief discussion and a CD containing a selection of open source tools that might or might not be a useful part of your Sarbanes-Oxley compliance plan. These tools are mentioned when they talk about the relevant parts of COBIT, but they aren't discussed in enough detail to help you decide whether they're the right tools for you.

As an example of compliance policy, they offer a password compliance policy that violates almost every rule for a good

password policy. It mixes information of interest only to administrators with information for users. It doesn't give the users an understandable reason for the policy. It states rules for passwords almost entirely in the negative ("Don't do . . .") and includes pointlessly specific rules. An editing error has caused the only useful information on picking a password to be attached to the "Enforcement" section. And there's no verification mentioned.

I'd pass this one up. Stick to Web resources and separate books on Sarbanes-Oxley and open source.

SOFTWARE PIRACY EXPOSED

Paul Craig

Syngress, 2005. 310 pages.
ISBN 1-93226-698-4

REVIEWED BY SAM STOVER

Since I'm not in on the piracy scene, I can't vouch for the technical accuracy of this book, nor can I just build a lab and put its assertions to the test. But what a fascinating read. I mean, this book had me hooked from page 1 to page 296 (right before the Index). Literally, I couldn't wait to get back to it after setting it down. I'm no stranger to BitTorrent, and we've all been hearing the media hype on Napster, Gnutella, etc., for years. When I first picked up this book, I expected to read about those very applications and their detriment to the Internet, and society as a whole.

What I found was an extremely detailed and thorough journey into the world of piracy, a world that most people don't know exists, much less interact with. Let's get one thing straight—P2P applications like eDonkey and BitTorrent are NOT piracy, at least not the piracy this book speaks to. Piracy is the high-adrenalin world of stealing or cracking applications and posting them to private sites. This world seems to be fueled by peer acceptance rather than monetary gain. Not that there isn't a an underlying "stick it to the man" attitude in the piracy groups, but as presented in this book, fiscal gain is not the primary motivation. Whether it's two couriers racing to get the same application distributed first, or the cracker pitting his sk1llz against the latest anti-piracy measures, it's all about competition.

Unlike other (dry) technical books, this one was extremely thought provoking. I still find myself discussing or contemplating the points the author brings to light. Throughout the first two-thirds of the book, I kept thinking "I can totally see why people get into this." Then I got to Chapter 9, where the high-profile FBI busts were discussed, complete with actual names and sentencing details. Then I started thinking, "Why would anyone do this?" Risking 10 years in prison for something that doesn't pay the bills seems a little extreme to me. A lot of the pirates seem to have day jobs, and piracy is more of a hobby/passion than a career. And some of the achievements are just astounding. Disk storage is measured in Terabytes, bandwidth is FastEthernet or even GigE, and the number of applications distributed in the thousands. Wow. If only the dot-bomb businesses had been this efficient.

The book has a lot of facts and plenty of interviews with real pirates. The research seems very sound, and the interviews ring true. Each aspect of the piracy scene is discussed in depth, from the Suppliers, to the Crackers, to the Distribution Chain. I found the technology discussed in the Cracking section especially interesting, as the author goes into a fair bit of detail when describing common reverse-engineering methods.

The blurb on the front cover bills this book as a "Must Read for Programmers, Law Enforcement, and Security Professionals." I agree totally. In fact, I think it should be required reading, especially for application developers, because if you code something, someone is going to pirate it. You need to know how and why.

My only complaint was the number of editorial oversights in the book. Misspellings and grammatical errors kept popping up. As with some other Syngress books I've read, I'd say that this was rushed to press because they thought the content was ground-breaking. Well, I agree. Just an amazingly fun read.

OS X FOR HACKERS AT HEART

Ken Caruso, Chris Hurley, Johnny Long, Preston Norvell, Tom Owad, Bruce Potter

Syngress, 2005. 439 pages.
ISBN 1-59749-040-7

REVIEWED BY SAM STOVER

A lot of folks are (or consider themselves) "Apple bigots." I tend to prefer the label "OS X bigot," but after reading this book, I'm starting to convert. Having only used OS X for about three years, and never once with a classic application, I think of OS X as "*NIX that works" or "*NIX for the masses" or "*NIX that's so freaking sexy I can't believe it." Take your pick.

I knew that Snort, Nessus, and KisMAC worked just fine. I knew you could integrate a Mac into a predominantly Windows-biased environment via SMB support, Entourage (the Mac version of Outlook), and Open Directory (the Mac version of Active Directory). I knew I

could compile my own source code manually, or use Fink and/or DarwinPorts for a more automated experience. I knew that most/all of these issues were in this book, and figured there wouldn't be much left for me to take home.

I knew nothing.

I didn't know that my Powerbook knows when it's being dropped, and reacts accordingly by parking the hard drive head. In fact, it does more than that—it keeps a running three-dimensional profile of its position in space and monitors G-forces to determine when it should panic. Turn your Powerbook sideways and you can read it like a book. It knows.

I didn't know that I could run CD-based Linux distributions from VirtualPC. Want to give the new Helix or Auditor ISO a spin? Drop it into VirtualPC, and away you go. Obviously, this isn't a long-term solution, but it will do if your Linux box dies (which never happens, right?).

And, most important, I really knew nothing about the great stuff that long-time Mac users take for granted, like Automator and AppleScript. Sure, I've messed around with AppleScript every now and then, but I always end up going back to Python or possibly shell scripting to get things done. The chapter on getting the most out of combining Automator, AppleScript, and any other language (Python, bash, Perl, C, etc.) totally rocked my world. Apple really goes out of their way to make it easy for the user to find the best way to get things done, and this book is truly the hackers' cookbook for putting it all together.

I totally enjoyed this book and would recommend it to anyone who has picked up a Mac and wants to run it through its paces. I would also recommend it to anyone contemplating getting a Mac, because I guarantee you'll end up making the purchase after you start salivating over what it can do.

My only true gripe with this book is that the editing really needed more attention. There weren't many chapters that didn't have at least one error, with the top scorer containing 19. This tells me that Syngress really rushed this book through to get it out to me, and, well, to you too. I suspect the 2nd edition will be a bit cleaner, but don't wait for it. If you want to learn what you can do with a Mac, you need this book—warts and all.

ESSENTIAL PHP SECURITY

Chris Shiflett

O'Reilly Media, 2006. 109 pages.
ISBN 0-596-00656-X

REVIEWED BY RIK FARROW

PHP is a very popular language for creating Web scripts, and one with a bad reputation for security. Shiflett argues that much of this reputation is undeserved, and the issues can be avoided by carefully following a set of principles when writing with PHP. I agree, to some degree.

This little book is an excellent way to learn about the security pitfalls one may encounter, and defend against, when writing Web scripts in any language. By following all of Shiflett's recommendations, you would avoid most, if not all, security vulnerabilities in PHP. If you use PHP, I highly recommend that you get this book, read it, and adhere to the suggestions found within it.

My only reservation is that I prefer languages that make it more difficult, if not impossible, to do the wrong thing. PHP lets you shoot yourself in the foot so many ways, that caution becomes the watchword.

UNDERSTANDING THE LINUX KERNEL, THIRD EDITION

Daniel P. Bovet and Marco Cesati

O'Reilly Media, 2006. 923 pages.
ISBN 0-596-00565-2

REVIEWED BY RIK FARROW

I really didn't want to understand the Linux kernel. Operating system programming is difficult, the Linux kernel is immense, and I have other things I must focus on. But when I found myself having to tinker with the kernel, or interested in learning about how modern memory management with 80x86 CPUs works, I needed a reference that could help me. And *Understanding the Linux Kernel* really worked for me.

Explaining a program that is millions of lines of code long is an enormous challenge. This book focuses on the operating system aspects of the kernel, as opposed to networking or device drivers (which are covered in other books). Given that focus, I feel that the authors have done an excellent job. They take the time to explain the issues clearly, and they provide cross-references to other areas of the book (and the kernel).

This was not the first Linux kernel book that I looked at, but it is the one I can recommend.