;login:

THE MAGAZINE OF USENIX & SAGE

August 2002 volume 27 • number 4

inside:

BOOK REVIEWS

Salus: The Bookworm

Harding: Review of Brian Shea's Have You Locked the Castle Gates

Karhuluoma: Review of Viega & McGraw's Building Secure Software

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

the bookworm

by Peter H. Salus

Peter H. Salus is a member of the ACM, the Early English Text Society, and the Trollope Society, and is a life member of the American Oriental Society. He is Chief Knowledge Officer at Matrix NewSystems. He owns neither a dog nor a cat.

peter@matrix.net

BOOKS REVIEWED IN THIS COLUMN

THE LANGUAGES OF EDISON'S LIGHT
CHARLES BAZERMAN
Cambridge, MA: MIT Press, 2002. Pp. 416.
ISBN 0-262-52326-4.

BEOWULF CLUSTER: COMPUTING WITH LINUX
THOMAS STERLING, ED.,
Cambridge, MA: MIT Press, 2001. Pp. 496.
ISBN 0-262-69274-0.

JIM SATO'S LEGO MINDSTORMS: THE MASTER'S TECHNIQUE
San Francisco: No Starch, 2002. Pp. 364.
ISBN 1-886411-56-5.

MAKING PROCESS IMPROVEMENT WORK
NEIL S. POTTER & MARY E. SAKRY
Boston: Addison-Wesley, 2002. Pp. 169.
ISBN 0-201-77577-8.

REQUIREMENTS BY COLLABORATION
ELLEN GOTTESDIENER
Boston: Addison-Wesley, 2002. Pp. 333.
ISBN 0-201-78606.

IP ROUTING
RAVI MALHOTRA
Sebastopol, CA: O'Reilly & Associates, 2002.
Pp. 219.
ISBN 0-596-00275-0.

TCP/IP NETWORK ADMINISTRATION, 3RD ED.
CRAIG HUNT
Sebastopol, CA: O'Reilly & Associates, 2002.
Pp. 730.
ISBN 0-596-00297-1.

I was going to devote this column to recent books on SQL, but I've spent the last four months traveling: in Scandinavia, Brazil, and the Netherlands (as well as the US). During this time a number of queries and interesting books have appeared, so I'm putting SQL off for the future.

I also want to express formal thanks to the folks running NORDU2002, to Michael Hejlskov Jacobsen and Sonny Larsen in Aarhus, to Gabriela Conceicao of the Sociedade Brasiliera de Computacao, to all those involved with the Forum Internacional Software Livre in Porto Alegre, and to Marielle Klatten and all the others who participated in SANE '02.

I was quite overwhelmed in Porto Alegre. There were 3500-plus at what I now understand is the largest software event in Latin America. I learned that the Brazilian state of Rio Grande do Sul has adopted open software and that the state of Minas Gerais was in the process of doing so. An anti-proprietary software bill has been introduced in Peru, too. The use of Linux and FreeBSD is amazingly widespread.

Interestingly, I seem to get asked many questions about the history of technology, not merely computing, so I'm going to begin with a history book.

Lux Fiat!

MIT Press has come out with a paperback edition of Bazerman's fascinating book on Edison. I recommend it to all of you. This is not a book about inventing electric light. It is about business and dealing with the public, with financiers (VCs?), and with government agencies. Bazerman is interested in symbols and sociology, and the electric light has become an incandescent and illuminating symbol.

Epic Poetry

Beowulf is the earliest epic poem in English, comprising a large part of the British Library MS. Cotton Vitellius A.xv. The use of Beowulf in computing has nothing at all to do with a mythic Germanic hero. *Beowulf Cluster: Computing with Linux* is a first-rate collection of essays supplying readers with a range of excellent tools for assembling a Beowulf cluster and for resource management.

LEGOs that Walk

Jin Sato is a major inventor. In *Jim Sato's LEGO Mindstorms*, there are toys and tricks to entertain nearly anyone. I was just thrilled to read about and examine the photos and diagrams of the robotic toys – are they really toys? – that Sato has developed. The chapter on "Making Mibo Walk" was really fascinating, if only because I can remember Marc Donner's brilliant walking robot (1983!) and his doctoral dissertation. I visited a lab at the University of Aarhus where LEGO robots were running quite impressively.

Management and Collaboration

Potter and Sakry have produced a small, easily read volume which I zipped through on a three-hour flight. *Making Process Improvement Work* has a concise approach to software process improvement, eschews a lot of the psychobabble and jargon that many books contain, and is full of good examples and concrete steps you can take. This may be a

book reviews

way of getting rid of those chronic software management problems.

Ellen Gottesdiener's somewhat larger (just about double the size) *Requirements by Collaboration* is about the human side of technology. More specifically, it's about how the quality of the various aspects of communication in a project, and most especially those employed when defining user requirements, strongly influence the success or failure of the project.

This isn't a book on building the product appropriately: it's a book about selecting the product to build.

Networking

If you really want to understand routing, I recommend Radia Perlman's *Interconnections* (Addison-Wesley, 1999), but for a smaller, more current view, I can't imagine a better volume than Malhotra's *IP Routing*. He assumes some knowledge of networking, of IP, etc., but he supplies the information that I imagine every network administrator needs.

I was really pleased to see the new third edition of Hunt's book on TCP/IP administration. Even if you've read the half-dozen or so parts of Comer and of Stevens, you need a contemporary handbook. Though it has gained a lot of weight over the years, it's still a fine piece of work.

HAVE YOU LOCKED THE CASTLE GATE?

BRIAN SHEA

Boston: Addison-Wesley, 2002. Pp. 193.
ISBN 0-201-71955-X.

REVIEWED BY CHUCK HARDIN
chardin@suchdamage.org

This is an insecurity book. If you follow its recommendations, you will be less secure in several respects than you were before. I'm not referring to merely running Windows; I assume the people who use the advice in this book start with default Windows installations with lots of viruses and exploits. The advice in this book can actually make you less secure than that.

The book incorporates a tedious analogy between a homestead-cum-village and a computer system in the evolution of their respective defenses. The concept is flawed, but it might have been excusable if the author had written a sufficiently compelling narrative to draw the reader through the book. He did not. The narrative is deadly dull and won't give the reader any real insights into the problems with computer security. It reads almost condescendingly, as if the reader herself were the peasant described in the narrative, to be led by the hand through the necessities of defending a computer.

Much worse, however, is the danger that the reader will buy into the author's analogy and adopt his security model, which was well-described by the IETF's Site Security Handbook (*http://www.ietf.org/rfc/rfc2196.txt*) as "the theory of a hard 'crunchy' shell and a soft 'squishy' middle." Shea emphasizes firewalls and physical security but ignores secure transport over the network for many kinds of data. The problem with pursuing this castle-like model has been well explored by the security community: once your single line of defense has been breached, you're wide open.

Shea's advice for defeating spammers is not much better. He recommends clicking on any provided link to be removed from the list. That's a great idea...for the spammers. Yes, the author warns you that you might have just verified the validity of your email address to the spammer, but offers no advice to avoid that problem. His advice is rather worse than useless.

He also presents important issues in illogical and inconsistent ways. One example is his Table 1-3, "Risk Numbers and Descriptions." It will badly confuse inexpert readers; more experienced readers will be annoyed that the author mixes attack mechanisms and attack goals with no apparent recognition of the difference. A virus is an attack mechanism, malicious destruction of data is an attack goal, but they should not be distinct items on the list — many viruses accomplish malicious destruction of data. Yet there they both are.

Roger Grimes' *Malicious Mobile Code* (O'Reilly, 2001) did much of what this book does, and does it much better. Buy that and skip this. It's horrible horrible horrible.

BUILDING SECURE SOFTWARE

JOHN VIEGA AND GARY MCGRAW

Boston, MA: Addison-Wesley, 2001. Pp. 528.
ISBN 0-201-72152-X.

REVIEWED BY NIINA KARHULUOMA
niina.karhuluoma@nokia.com

Software without security is a huge mistake. It is extremely easy to produce a buffer overflow type of attack on a system, and this could just as easily be stopped by taking some important design principles into account while programming.

Security is much more than just techniques or protocols like IPSec, SSL/TLS, and firewalls. It is an integrated process, which must be remembered in every phase of developing a system.

book reviews

Building Secure Software is one of the few books dedicated to examining the production or choice of software with security features.

In general, the book is excellent reading for anyone who wants to learn about, or who is already working with, software-related security issues. The book is understandable and well written – it can be read without deep security knowledge.

The book begins with an introduction to security awareness and gives some basic points as to where to get information about security vulnerabilities as well as describing the goals of security. These goals are useful to anyone who needs to know the different levels of security and why they are needed.

Entire books could be written on software risk management, but the writers manage fairly thorough coverage in their chapter on this broad issue. The chapter makes clear the importance of incorporating security into the software developing process from the beginning.

This book also takes up the pros and cons of open source versus closed source software. The best part of this chapter is that it provides guidance about the issues that should be taken into account in making this decision but does so without advocating for one side or the other.

The "Guiding Principles for Software Security" section supplies 10 important guidelines that can be used to avoid most potential problems in the context of software security. These guidelines are useful although they need to be considered case by case because enterprises frequently have restrictions on software features. Different kinds of rules (e.g., those applicable to customers) may cause difficulties in applying these guidelines literally. Still, the principles can help software developers to achieve more secure products.

Viega and McGraw take a close (and quite broad) look at software auditing. Auditing is an essential method of checking code's features and security. This should not be done merely by using code review. This book introduces some general baselines of how to audit software and what kind of tools to use, and it also provides some views about the effectiveness of auditing. The writers have a healthy attitude toward code and software checking – they are really pointing out that auditing is a method that needs to be part of the software process.

Viega and McGraw take a thoughtful look at security aspects that need to be considered in software design. Issues like buffer overflows, access control, race conditions, randomness, and determinism are described extensively in this book.

My own background is in cryptography research, and I was very pleased to see a book that describes secure software also considering cryptography-related programming issues like randomness (including views on the handling of entropy). This makes the book very good reading for anyone who needs to write code for cryptographic purposes. This area is very seldom described and quite often programmers end up learning by doing.

The cryptographic part of the book is extensive and provides an excellent overview, especially if the programmer is using some other, cryptography-specific, reference at the same time. The book also includes a short section about cryptography basics and can be used as an overview of the most essential issues concerning cryptographic software creation.

This book is excellent – useful for teaching at the university level and serving well as a handbook for those specialists whose main job is to develop software with security features. This book is clearly written and is a good introduction for people who want to get an overview of security in the software process.