;login:

THE MAGAZINE OF USENIX & SAGE

April 2004 • volume 29 • number 2

inside:

USENIX

The Advanced Computing Systems Association

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 owns neither a dog nor a cat.

peter@netpedant.com

BOOKS REVIEWED IN THIS COLUMN

SENDMAIL COOKBOOK
CRAIG HUNT
Sebastopol, CA: O'Reilly, 2004. Pp. 388.
ISBN 0-596-00471-0.

POSTFIX
KYLE D. DENT
Sebastopol, CA: O'Reilly, 2004. Pp. 264.
ISBN 0-596-00212-2.

DESIGN RESEARCH
BRENDA LAUREL, ED.
Cambridge, MA: MIT Press, 2004. Pp. 334.
ISBN 0-262-12263-4.

XML HANDBOOK, 5TH ED.
CHARLES F. GOLDFARB AND PAUL PRESCOD
Upper Saddle River, NJ: Prentice Hall, 2003. Pp. 1200 + 2 CD-ROMs.
ISBN 0-13-049765-7.

I've got to begin with a confession this month: I was going to write about *Malware*. But Chuvakin's review came in, and it's one with which I concur 100%. So I've not bothered. Just read the review following this column and go with what Anton says. It's a fine book.

I also need to apologize: In the December column, I mentioned Waldrop. Knowing just how many computer folk read SF, I made an unwarranted assumption. Howard Waldrop is an SF author. I think he's up there with the best, but he's neither as famous nor as popular as Gibson or Sterling. So I apologize to those who wrote me and to others – I didn't intend to be a snob. (But if you get a chance, read some of Waldrop's stuff.)

We've All Got Mail

OK. We've all got copies of Costales, Allman, and Rickert (*Sendmail*), yet we still may not have completely mastered the sendmail.cf file (sometimes I wonder whether anyone has, but that merely exposes one of my shortcomings).

Craig Hunt's *Sendmail Cookbook* really helps. I was especially impressed by his chapter on AUTH (pp. 242–273); it is extraordinary. The chapters on masquerading (pp. 103–150) and securing mail transport (pp. 274–317) are very fine, too. While the chapter on spam is good, I fear that it just isn't enough. Unfortunately, I don't have a panacea. I get a lot of trash every day, and I can't adjust my filters as rapidly as header variants and bogus subject lines originate.

Craig, this is a fine job.

Another fine job is Dent's *Postfix*. This is a very good guide to a splendid MTA. Written by Wietse Venema while he was at IBM Research, Postfix was released as open source in 1998 and has become fairly widely employed since. This book is, to the best of my knowledge, only the second book on Postfix, and the less said about Blum's padded book of several years ago, the better.

Dent has done a really neat job. In fact, his chapter on blocking spam is a great supplement to Hunt's.

Designed to . . .

Design Research is a brilliant anthology, full of interesting articles and thought-provoking assertions. I found it extraordinarily difficult to read, however. This is because the book designers ("The Offices of Anne Burdick, Los Angeles") have run amok. Each section is demarcated by color (orange for Section 1; sage for 2; pale yellow-green for 3; etc.). The yellow-green was near impossible for me to read. I have no idea what the diagram labels on p. 143 say, or most of the headings in the following section; I cannot discriminate the letters from the background.

So, in some manner, Section 3 ("Process") was opaque to me. Many of the texts were interesting, even though they were periodically interrupted by areas of color in which some things were intelligible, but the whole was not.

The volume is "over-designed" to death, resembling an early issue of *Wired*. Surely, this is not the result of any sensible "design research."

I have long admired the work of Brenda Laurel. What a pity to have the content marred and obscured by out-of-control designers.

XML

There's a new (fifth) edition of Goldfarb's *XML Handbook*. Previous editions were quite useful. The new one is quite enormous, and could use a bit of editing and some reorganization. This is an excellent updating, nonetheless.

book reviews

MALWARE: FIGHTING MALICIOUS CODE

ED SKOUDIS (WITH LARRY ZELTSER)

Upper Saddle River, NJ: Prentice Hall, 2004.
Pp. 647.
ISBN 0-13-101405-6.

Reviewed by Anton Chuvakin

I rarely label something a "masterpiece," but Ed Skoudis' *Malware: Fighting Malicious Code* is nothing short of that. The book is an amazing combination of depth and breadth, which I always love in a security book. Moreover, it combines these with lively and easy to follow presentation style as well as Ed's trademark humor (featuring the traditional overuse of the word "evil"). In many regards, the book is more fun to read and more packed with material than his previous work, *Counterhack*. The book also strongly conveys the excitement that the author obviously feels about this field.

The book covers the wide scope of malicious code (viruses, worms, mobile code, rootkits, Trojans, backdoors) in a logical and well-structured fashion. This is not your grandmother's "virus book," as it covers all sorts of malicious programming and scripting. Chapter summaries, reasons "why you need to know," examples, clear diagrams, accurate analogies (often abused in other security books) are all there to educate and entertain. Early on, I thought that some of the examples were a bit simplistic, but later I noticed that they worked extremely well, especially for some of the technologies I was not intimately familiar with (such as the Windows kernel).

The book starts with a nice, clear definition of "malicious code," which helps to set the frame for the rest of the book. It goes on to cover all the types of malware outlined above. Highlights included the exciting material on future worms and possible trends in worm activity; coverage of various browser-based attacks, including evil plugins, ActiveX, and XSS (as utilized by malware); the presentation on sniffing backdoors and hacks using VNC; and the coverage of source-Trojans (with detailed analysis of recent attacks against common open source software) and some neat data-hiding tricks.

The section on rootkits (two chapters for application and kernel-level), however, was my favorite, presenting this malicious technology in a logical, very well-written fashion. Starting with brief but useful overviews of Linux and Windows kernels, coverage continues by noting "five ways to manipulate a kernel" for malicious purpose. The material on Windows rootkits and kernel tricks is fascinating. Several examples of fairly recent kernel rootkits are analyzed for both platforms.

If the rest of the book is exciting, the author's discussion in Chapter 9 of the possibility for BIOS and CPU microcode malware is simply awesome. The book follows this up with coverage of some end-to-end malware-related attacks scenarios, which are lots of fun to read.

The book is topped with a chapter on analyzing malware, complete with suggestions for a lab setup and a structured presentation of various analysis approaches (static and dynamic). An analysis template is there as well.

Overall, the book is a great read for any security professional, system admin, or aspiring hacker. Its focus includes both attacks and defenses, with a slight bias toward attack (it also often touches on "defenses against defense" tricks, utilized by malicious software). UNIX and Windows platforms are both covered at almost equal levels of detail.

SECURITY WARRIOR

CYRUS PEIKARI AND ANTON CHUVAKIN

Sebastopol: O'Reilly, 2004. Pp. 525.
ISBN 0-596-00545-8.

Reviewed by Rik Farrow

Security Warrior is touted as an advanced book, and some parts of it actually are. I obtained a copy of the book because I was interested in learning more about reverse engineering of hostile code. The book does start out with four chapters on reverse engineering, with the chapter on working with Linux the most extensive in terms of material and explanation.

The second chapter covers Windows code disassembly tools and says, quite correctly, that since access to source code is rare in the Windows world, the tools have considerably greater maturity than in the Linux world. The authors mention several of the tools, but do not discuss the structure of Windows programs, which really disappointed me since I wanted to learn more about Windows disassembly. Chapter 3 goes into much greater detail about the structure of Linux programs and how the C compiler works. My impression was that the authors assume that their audience already understands Windows programming in intimate detail, and they themselves are exploring how to disassemble Linux with the primitive tools available.

The authors do provide working code and scripts that help with disassembling Linux programs, and that is a real plus. These code examples can be found on one of the authors' Web site. But the real focus of the disassembly techniques does not appear to be exploring hostile code, but discovering how to bypass checks on serial numbers and other copy protection or access control schemes. That was not what I wanted, as I expect to see more hostile Linux code to appear in the future. I expect code that, like the viruses and worms familiar from its Windows' counterparts, will not come with source code.

The remainder of the book provides a beginner to intermediate text on general computer-security topics, with some glaring errors. For example, on page 186, at the end of the TCP/IP handshake, the "command is received and resets the sequence number to zero." Huh? Haven't these guys spent any time with their eyes glued to sniffers?

The chapter on UNIX security touches on some interesting topics, but provides little useful advice. The suggestion that some accounts in the passwd file should simply be deleted appears seriously misadvised, and the authors completely miss the significance of ownership and permissions on system directories – pretty basic stuff.

This is not a bad book – I simply wish that it had been better tech edited and more focused on reverse engineering of hostile code. There are other books that cover incident response, honeypots, UNIX security, and other topics in much greater detail.

QUICKSILVER

Neal Stephenson

San Francisco: Harper Collins, 2003. Pp. 926.
ISBN 0-380-97742-7.

Reviewed by Rik Farrow

While not a technical book, *Quicksilver* seems to me worth mentioning since it was written by Neal Stephenson, the keynote speaker at USENIX '03 and author of *Cryptonomicon*, a very popular book among the computing community. When I learned that Stephenson planned on venturing into historical fiction, I was at first disappointed. But *Quicksilver* did not disappoint me. It is a book to savor.

Quicksilver deals with the people and events at the end of the 16th century, primarily in England and Europe. Throughout the book, Stephenson brings to life characters such as Newton, Leibnitz, Hooke, and other famous natural philosophers in a way that makes the era in which they lived exciting and real. Stephenson examines not only the birth of science, but the political, religious, technological, and social structure on which it depended. Stephenson writes superb dialogue that reveals his characters' thoughts and feelings while educating the reader at the same time. I loved his exploration of the evolution of the financial world of banks and markets.

The book's pace is somewhat uneven in that there are slow sections – for example, exchanges of letters. But these are more than compensated by the brilliant episodes of action salted throughout the book. I found myself reading the book for its prose and dialogue, learning about this period of history, and then getting caught up in a chase scene that just could not be postponed. I highly recommend this book, both for its entertainment value and for what you can learn not only about history but also about how humans operate.