book reviews

ELIZABETH ZWICKY

zwicky@greatcircle.com

with Sam Stover, Heison Chak, and Rik Farrow

MIND PERFORMANCE HACKS: TIPS & TOOLS FOR OVERCLOCKING YOUR BRAIN

Ron Hale-Evans

O'Reilly, 2006. 308 pages.
ISBN 0-596-10153-8.

This is a nice book, with a lot of good advice in it. I made the mistake of recommending it to my husband, and then had to pry it out of his hands to review it. Because it contains a lot of disparate stuff, different parts of it are going to appeal to different people. For instance, the author puts an early emphasis on memorization tricks, which I think are flashy but not of much use in daily life, but then he gets into what I think of as the good stuff: managing the information in your life, creativity, decision making, communication, mental clarity. If you are, for instance, a medical student, then memorization tricks may actually be good stuff from your point of view. There are also a bunch of math tricks, some of which I find handy (including a couple I didn't know) and some of which are too much work for the amount of math I do. But what I think is amusing but impractical might be exactly what gets you going.

There is one hack where I think Hale-Evans got it 90% wrong. Hack #74 quite correctly recommends Karen Pryor's marvelous book *Don't Shoot the Dog*. It then promptly turns around and recommends the two least effective methods of self-training available: bribery and punishment. This is so wrong-headed as to be just silly. Pick up *Don't Shoot the Dog* for an explanation of why these are ineffective and what you should do instead. And yes, she explicitly addresses self-training, both teaching yourself new habits and getting rid of bad habits.

I also think that you'd be better advised to seek out a foreign language than an artificial language. Artificial-language writers tend to go for only minorly mind-blowing moves, whereas an in-depth study of a foreign language gets you all sorts of subtle and not-so-subtle changes and access to an entire culture. And a broad study of foreign languages will get you into things like Polynesian pronouns and Swahili or Navaho noun classifiers, which will give you novel ways of breaking up the world. The regularization that goes into artificial languages removes some of the best parts of language study. If you want to make up your own language, you should definitely go study half-a-dozen real languages from different language families first. (Yeah, that is a lot of work. I guess we can tell where my heart is—I'm willing to learn foreign languages for amusement but not finger arithmetic.)

PERFECT PASSWORDS: SELECTION, PROTECTION, AUTHENTICATION

Mark Burnett

Syngress, 2006. 178 pages.
ISBN 1-59749-041-5.

There are a few nice, novel ideas here, and some clear explanations of important concepts (you'd be surprised how many people think a 14-character password is twice as good as a 7-character password). Most of it is not going to be news to experienced administrators, but Burnett's book would be a great gift for people just encountering the password issue, and there are a few lovely ideas, of the "simple but life-changing" form. When you run across particularly annoying and counterproductive password policies, you could force a copy of this book on the authors.

That said, I found it hard to love. I think I would have liked it better if it were shorter; it began to feel repetitive pretty fast. If you're an administrator, it will give you good ideas about what sort of password policy you should have, and what sort of passwords you, personally, should pick, but it offers scant help on communicating a password policy to users, enforcing it, or even making it possible on systems designed for short passwords. And it really, really does focus on what to do when you're faced with a reusable, text password; there's minimal discussion of other options. OK, so that's what it says it's about, but surely a mention of one-time passwords wouldn't go amiss instead of several pages of random character strings?

SOFTWARE SECURITY: BUILDING SECURITY IN

Gary McGraw

Addison-Wesley, 2006. 406 pages. ISBN 0-321-35670-5.

You are a software security person. You are surrounded by crazy developers, who say things like "But what does it matter what encryption algorithm we use?" and "Well, yeah, that would be bad. But nobody would ever do that." Or, perhaps, you are trying to build something out of pieces you strongly suspect were built by such developers. This book is meant for you. It gives you the tools to explain what is wrong and why it's wrong, and if you are part of the development process, the tools to get it right.

If you are a developer who wants to find out about this security stuff, this book will probably work for you too. It certainly tries to explain the issues to developers who don't understand them, but I'm not certain how convincing it will be to somebody who's security-naive or security-hostile.

In general, I like this book a lot, for giving a general explanation of security in the software development process. The weakest section is chapter 2, where it explains a risk management framework. There's lots of supporting verbiage about how difficult the material is, but I came to the conclusion that the material itself isn't particularly mind-boggling; it just hasn't been freed from business-ese enough to make it palatable, so there are lots of sentences such as, "Management of risks, including the notion of risk aversion and technical tradeoffs, is deeply impacted by business motivation." Given that the author can write lucidly about difficult security concepts, either there is a strong effect of a previous author or the author suffers from a common speech impediment, in which a business context causes a sudden inability to communicate rationally.

LINUX DEBUGGING AND PERFORMANCE TUNING TIPS AND TECHNIQUES

Steve Best

Prentice Hall, 2005. 427 pages.
ISBN 0-13-149247-0.

Since I ended up with a bunch of Linux performance tuning books in my first review batch, I've become curious about Linux performance tuning books. This one is very straightforward, mostly talking about the nuts and bolts of how to use the tools—the command line options, what the output looks like, how to set up your situation so you can use a tool. It covers a wide variety of tools and gives good examples of how they can be used and how you interpret the results.

If you have a specific problem to solve, and a general introductory-level understanding of performance tuning and debugging, and you want to know what Linux tools are available and how to use them, this is a good tool for that purpose. It is not something you would want to read from end to end; it's more of a reference work. If you want a general education on Linux performance, Ezolt's *Optimizing Linux Performance* is still the way to go (see my review in October 2005).

PENETRATION TESTER'S OPEN SOURCE TOOLKIT

Johnny Long, Aaron W. Bayles, James C. Foster, Chris Hurley Vincent Liu, Mike Petruzzi, Noam Rathaus, SensePost, and Mark Wolfgang

Syngress, 2006. 704 pages.
ISBN 1-59749-021-0.

Reviewed by Sam Stover

I'd like to clarify a critical point about this book: It is not just a collection of howto's for all of the programs included on the Auditor LiveCD distribution. It is a step-by-step guide into the many facets of penetration testing which uses the Auditor LiveCD to provide most of the tools needed.

If you are looking for a quick reference on how to use every tool included on the Auditor LiveCD, this book will disappoint you, and, honestly, it should. You should be Googling for that. If, however, you would like to learn more about a particular area of pentesting, or the discipline as a whole, you will love this book.

The first two chapters start out of the gate with reconnaissance and enumeration. From there the chapters become a bit more application-specific, focusing on databases, Web servers, wireless networks, and, finally, network infrastructure devices. The last seven chapters deal with either writing your own pentesting tools or using the two most ubiquitous pentesting frameworks, Nessus and Metasploit.

I found that this book worked great as a reference for areas where my knowledge was lacking. For example, I haven't spent much time pentesting databases. So I turned to page 149 and dove in. The chapter was clean, easy to read, and to the point. There were tips for Oracle and Microsoft databases, as well as suggestions for how to make a database more secure.

Nessus and Metasploit get a fair degree of special attention throughout the book, as well they should. Not only are there chapters dedicated to each (four chapters on Nessus and two on Metasploit), but they are also discussed when appropriate in the other chapters. In the database chapter, there is a section on the Nessus database checks tucked between OScanner and SQLAT.

One thing that budding pentesters fail to realize is that the real value in a pentest is not in pointing out the deficiencies but in making suggestions on how to fix them. This book gives you both sides of the equation, which also means that this book should be on the bookshelf of any system, security, or network admin. If you are responsible for a Web farm, why not use the same tips and tricks that the pentesters are using? You don't even have to go out and find the tools: They're already on the included Auditor LiveCD.

In short, I think this book should appeal to a wide and varied audience. Experienced pentesters probably won't find anything new here, but people looking to jump into the industry, as well as any admin, will find this book to be a easy and fun introduction into the mentality and tools of penetration testing.

VOIP HACKS

Ted Wallingford

O'Reilly, 2006. 306 pages.
ISBN 0-596-10133-3.

Reviewed by Heison Chak

As I was upgrading my Asterisk PBX server to the latest release, I started flipping through the pages of *VoIP Hacks*, hoping to get some inspirations from the 100 Internet Telephony tips and tools.

I found most of the hacks clearly written, with enough examples to explicate the descriptions, and there is a good balance among tools that could run on Linux, Windows, and Mac OS X. Hacks on how to use an Intel V.92 Winmodem card to replicate the Digium X100P FXO card (which is no longer carried by Digium) and intercepting a VoIP call on switched networks using ARP poisoning may be a little controversial, but isn't that what hacking is all about?

There were a few times when I wished there had been a little more detail. For example, when the book described examples of building a fax-to-email gateway using spandsp, an example on how to build an email-to-fax gateway could be the very next question on a reader's mind.

I really appreciate the work that went into the hacks; it recalls memories of the drawing board when I was building my VoIP environment with Asterisk. It is definitely a book for the beginner-to-intermediate VoIP enthusiast. Experts may find a lot of the ideas very familiar.

LINUX PATCH MANAGEMENT

Michael Jang

Prentice Hall, 2006. 262 pages.
ISBN 0-13-236675-4.

Reviewed by Rik Farrow

This is a book I wish I had had years ago, when I was tasked with creating a course about UNIX patch management. I found myself confounded with multiple versions of Linux, each with its own peculiar patching software. Jang does a fine job of covering all the major Linux distros, and some smaller ones as well.

Jang splits his focus between using and configuring individual tools, such as apt, yum, and YaST, and explaining how to set up local repositories of patches. Local repositories are important, not just to avoid beating up on your own network connection and the bandwidth of patch servers. Jang covers these issues well, and in enough detail, that you should be able to follow his instructions and set up your own patch repositories.

Jang does not deal with other issues involved in patch management, such as patch testing, reference systems, test deployments of patches, or staggering deployments, but focuses solely on the use of the tools. You can use this book to choose a Linux distro based upon the choice of patch management systems, as well as to support patching your existing Linux systems. I plan on keeping this book handy.

BUILDING EXTREME PCS

Ben Hardwidge

O'Reilly, 2006. 192 pages.
ISBN 0-596-10136-8.

Reviewed by Rik Farrow

This folio-sized book represents a departure from the usual run of O'Reilly products. Beautifully illustrated (in a very geeky sense) with full-color photos of cases, CPUs, water cooling systems, and more, Hardwidge's book takes you on a journey into the world of building extreme PCs that I believe will actually be useful for anyone building their own PC. Why? Because some of his tips will be useful to those who have chosen to DIY instead of buying the latest off-the-shelf clone PC.

Hardwidge includes discussions of all the key PC components. I found his primer on current CPU technologies very helpful, for example, as he explains the difference between current Intel and AMD offerings, differences in cache types, cache levels, etc. None of the explanations is very deep, but this may be exactly what you need when you want answers in a hurry. You might wonder why not Google for this, and I have tried, but I have found much better answers here.

Hardwidge really targets people building Windows systems for gaming, but he also includes silent PCs and PCs suitable for PVRs. *Building Extreme PCs* is almost a coffee-table book in the quality of the illustrations, depending on your taste (or your significant other's feelings about tech in the living room).