

book reviews



ELIZABETH ZWICKY, WITH JASON DUSEK, EVAN TERAN, AND RIK FARROW

THE GREENING OF IT: HOW COMPANIES CAN MAKE A DIFFERENCE FOR THE ENVIRONMENT

John Lamb

IBM Press, 2009. 304 pages.
ISBN 978-0-13-715083-0

There's no doubt that green is in and that companies have begun to notice how computers actually do use up resources—they may not produce black smoke, but they sure do spend a lot of time converting expensive electricity into expensive and annoying heat. And, from an IT perspective, many of the things that go into being greener are easy to verbalize but hard to do.

Anybody can see what the three major goals are: use fewer computers (turn off unused machines, use machines more efficiently to create more unused machines), optimize cooling, and use newer, more efficient computers. Unfortunately, this requires venturing into a lot of dangerous territory. Consolidating services and replacing machines both require a great deal of management support. As for optimizing cooling, I may be projecting my feelings onto other people, but it seems to me that IT people regard HVAC people much the way that non-computer people regard computer people, as semi-trustworthy keepers of black arts. Those of us who remember the days of converting away from mainframes, when we removed giant heat-generating computers and the buildings we worked in never again cooled the right amount in the right places, may never again trust heating and cooling systems.

I was hoping that this book would make me feel more optimistic about these things, and in some ways it did. It has case studies showing that other people have navigated these treacherous waters, and it has a suggested process for achieving greener data centers. It also has some practical advice on things you can do. But that advice leaves a lot of gaps. Want to measure how well your cooling works and what your power is being used for? IBM has a solution for that. It sounds pretty cool, but it also doesn't sound cheap or readily available.

If you want an overview of green issues in IT you can share with your CIO and other managerial types, this is a reasonable choice. It covers government and power company green initiatives that you are not likely to find elsewhere, and it should encourage people to take reasonable steps. If you want a guide that will help you actually make the changes, this is not going to do much for you.

THE NIKON D90 COMPANION

Ben Long

O'Reilly and Associates, 2009. 273 pages.
ISBN 978-0-596-15987-0

I was asked once if somebody should read the D90 manual, to which I replied “No.” “Oh,” he said, “is it written for people who understand f-stops, then?” I thought about it for a moment and came to the conclusion that the D90 manual, like most camera manuals, was not written for an audience, but as a checklist. It tells you everything about each button and knob on the camera, and provides all the legal warnings you might ever want. A determined person with a good background in digital photography can figure out quite a few things from it, but not with any enjoyment except perhaps the feeling of having successfully defeated a challenge.

The D90 Companion is the book my friend was looking for; it assumes that your goal is to take good pictures and tells you about that in the context of the D90. It starts from basics, both about digital cameras and about photography in general, and takes you through learning the camera in a reasonable order. Its advice on photography in general is sensible, and it helps you understand what situations are appropriate for what settings. I'm tolerably familiar with Nikon digital cameras (the D90 is my second Nikon DSLR and my fifth Nikon digital camera), and I still learned things from it.

This book is best suited for somebody who's reasonably new to digital SLRs. I enjoyed it, but I didn't need it; its prime audience is people who have a D90 and are feeling either intimidated or frustrated, knowing that the camera can do lots of things and

finding themselves still turning it to auto and getting nice snapshots.

ALTERNATIVE DNS SERVERS: CHOICE AND DEPLOYMENT, AND OPTIONAL SQL/LDAP BACK-ENDS

Jan-Piet Mens

U/IT Cambridge, Ltd. 2009. 652 pages.
ISBN 978-0-9544529-9-5

If you are a reasonably experienced system administrator with a need to make DNS stand on its hind legs and dance, this is the book for you. In fact, it should be useful even for simpler DNS configurations, as long as you come at it with a good basic understanding of security and system administration. You should note that it does in fact cover BIND, although it emphasizes coverage of BIND's database back-end extensions.

This book covers a wide range of UNIX-based DNS servers and their database back-ends. There is a brief discussion of Microsoft Windows-based options and a somewhat more involved discussion of how to write your own trivial nameserver, why you might want to, and how you might add exotic features to your name service. You will also find monitoring and performance advice. While the book does review the basics of DNS and provide some advice about choosing DNS servers and designing a DNS infrastructure, it's aimed at the kind of people who are willing and able to write their own utilities. It provides a lot of facts and advice, but there's not much handholding going on here.

As a security person, I particularly noticed that there are no warning notices about configuration files that contain clear-text credentials for database accounts which may have write permission. (Protect these very well. They are dangerous.) Also, while the author mentions that you may want to ensure that name servers with full-powered database back-ends are not Internet-accessible, he assumes that you know why that is (databases and security do not go together like peanut butter and chocolate).

BAD SCIENCE

Ben Goldacre

Harper Perennial. 2009. 339 pages.
ISBN 978-0-00-728487-0

Mostly I review books because they're new, and you might want to know about them. Periodi-

cally I review a book simply because I love it and think you ought to hunt it down and expand your mind—in a technological way, of course (I promise never to inflict my taste in fiction on you). This is one of those books, and, worse yet, it's not in print in the US currently. Hunt down a copy somewhere, and if possible, hunt down this 2009 edition, because it has an extra chapter that was still under litigation in the first edition. And also seek out the author's blog at www.badsience.net.

Why should you, presumably some sort of computer professional, care? Ben Goldacre is a doctor, and he is primarily writing about things at least apparently related to medicine. But, in fact, his main themes are entirely relevant to technologists of all stripes, and they are:

- Doing science (real science, where you make hypotheses and test them) is easy, fun, and rewarding. Try it at home! At work! Wherever you are right now!
- Press coverage of sciency-stuff (which includes not only medicine, but also computer science) is terrible beyond belief. There are reasons for this, but still, it's unimaginably bad and it hurts people.
- Statistics is not that hard to understand and apply, particularly when it comes with something emotionally gripping. Sure, comics and pictures may help, but here your statistical education is enhanced with villains. Swindlers, cheats, the painfully misguided, and the insult- and lawsuit-throwing trolls all show up. If that, and the fact that the examples are about things that might kill you, doesn't grab your attention, nothing ever will.

BEAUTIFUL SECURITY: LEADING SECURITY EXPERTS EXPLAIN HOW THEY THINK

Andy Oram and John Viega, editors

O'Reilly, 2009. 268 pages.
ISBN 978-0-596-52748-8

If you're a security person, the very title *Beautiful Security* is enough to give you warm fuzzy feelings. That's because "security" may sound like a good thing, but for computer people, it's a source of many kinds of nastiness. People associate it with inconvenience, feelings of helplessness, and nasty people in black. The idea that people are trying to associate it with "beautiful," which is more about butterflies and pleasure, can only be a good thing.

And for the most part, this book is a good thing. It focuses on several important themes (security is an integral part of design, it's not an unsolvable problem, it's not just about the computers but also about the legal system, beauty in the computational sense is necessary for security) and often succeeds in com-

municating them clearly enough to get through to an interested but not particularly knowledgeable audience.

At the same time, it suffers from being an anthology of essays, and it suffers from attempting to be cutting-edge and accessible at the same time. I liked some of the essays a lot—the essay “Psychological Security Traps” that starts out the volume is clear and compelling—but some of them were badly edited (there are references to things that have been edited out in several of them), and I found the final essay completely unconvincing. It’s not the only product-oriented essay in the book, but it’s the only one that annoyed me; it does not succeed in making the case that the product’s technology is substantially new, but instead attacks existing solutions.

I’d recommend this book to somebody with a technical background who’s looking for something interesting about current security issues. It may also be useful for security-phobic managers since it is, on the whole, reassuring about security as a functioning part of an organization.

MASTERMINDS OF PROGRAMMING

Federico Biancuzzi and Shane Warden

O’Reilly, 2009, 494 pages.
ISBN 978-0-596-51517-1

REVIEWED BY JASON DUSEK

Masterminds of Programming presents conversations with the developers of many celebrated languages—a mix of old (AWK, ML) and new (C#, Python), classic (C++, BASIC) and esoteric (Haskell, Forth), ubiquitous (Perl, Java) and niche (Lua, APL).

The omission of a LISP is a disappointment. The interview with Milner on ML is an unexpected delight. We don’t get a C interview, but a lot of relevant material is covered in the AWK interview.

The chapter on AWK manages to cover a wide range of topics in computing: the role of documentation in project management, the “little languages” philosophy and compositionality in UNIX, types, the relationship between language-level modularity through object support and system-level modularity through tools. Naturally, these are mingled in with AWK specifics: AWK’s competitor at PARC, Bell Labs in the

seventies, the late adoption of comprehensive tests for the project.

The Objective-C interview is of similarly broad interest. It’s usual to associate Objective-C with Apple but its origins were in development of telecom systems; the interview thus presents a wealth of material on componentization, distributed work groups, and system evolution. The material from Brad Cox, in particular, moves straight into the relationship among various component models for software: SOA, Java’s JBI, and the more language-agnostic SCA.

All the interviews strike this balance between language-specific issues and those of general interest; the lessons of history are obscured neither with trivia nor with theory.

PYTHON FOR UNIX AND LINUX SYSTEM ADMINISTRATION

Noah Gift and Jeremy Jones

O’Reilly, 2008, 456 pages.
ISBN 978-0-596-51582-9

REVIEWED BY JASON DUSEK

This book is a good introduction to Python, starting with straightforward examples of the same construct in sh/Perl/Python and then moving on to the core language and interactive usage. A number of libraries/kits are discussed in the context of an operations team’s use thereof: SNMP, LDAP and DNS toolkits, networking, serialization, packaging are covered, among many other topics.

Overall, the book makes a solid case for Python’s place in the sysadmin’s toolkit; the book ensures you’ll not be at a loss when you need that one thing, whatever it is, and you know what it’s called in Perl.

GRAY HAT PYTHON

Justin Seitz

No Starch Press 2009. 189 pages.
ISBN 978-1-593-27192-3

REVIEWED BY EVAN TERAN

Python is an excellent language for reverse engineering; its only real drawback is the lack of a centralized source of information and examples. This book attempts to fill this gap and succeeds in covering what you need to know.

Chapter 1 walks you through things as simple as the process of installing Python and as important as understanding the ctypes module. If you are a C programmer, you may have guessed that the ctypes module just provides Python versions of the func-

tions in the C `<ctypes.h>` header, namely a few minor character classification functions. Far more critical than that, it is the glue that lets Python code perform system-level tasks. This module basically gives you the power of C within Python. You can create structures and unions which perfectly match their C counterparts. More importantly, it lets you resolve functions found in shared libraries and use them directly from your Python code. When Justin Seitz says `ctypes` is a hacker's best friend, he isn't kidding. I'm glad that he explains this nice and early in the book.

Once that's under your belt, it's time to talk about actually using and making debuggers. Chapter 2 goes into detail about how x86 debuggers work, explaining how you would conceptually go about implementing all of the different types of breakpoints and why they work the way they do. If you are already familiar with the x86 architecture and just want to jump into the Python aspect of things, this chapter isn't strictly necessary, but it is a good refresher.

Next, we get into the nitty-gritty: making a debugger using the Windows API. Every important API call is explained in detail, complete with example code. Believe it or not, by the end of Chapter 3 you have all the tools and knowledge to construct a functional debugger. There are a few minor details that get glossed over. For example, properly unsetting a breakpoint (so you can have your debugger resume) isn't really mentioned in detail. That's okay though, because in reality if you want to write a debugger in Python, you'll be using `PyDbg`. You may wonder why you just read 30 pages on how to write a debugger only to find out that Pedram Amini wrote an excellent framework that handles all of the little details for you. In the end, though, if you understand how a tool works, you'll do a better job at using the tool.

Useful tools and different hooking and fuzzing techniques are discussed in good detail in the later chapters, and so is, finally, `PyEMU`, a very cool x86 emulator written entirely in Python. It lets you execute and debug malware without fear of infection (since it is running on a virtual machine).

There are only a few things I wished had been done differently in the book. For example, the focus is very x86-centric, while 64-bit computing is making its way into the mainstream and is only going to get more popular over time.

Also, I would have liked more Linux-centric examples. While it's true that the concepts are the same, the `ptrace` API is very different from the Windows debugger API, and it would have been nice to see the book compare and contrast the two. Overall, though, this is a great book. It covers all of the things that you will need to start using Python as your primary reverse-engineering language.

THE MANGA GUIDE TO ELECTRICITY

Kazuhiro Fujitaki

No Starch Press, 2009. 232 pages.
ISBN 978-1-59327-197-8

REVIEWED BY RIK FARROW

I tried another experiment with a Manga book, and it worked out pretty well. This guide follows the usual Manga format, in which a teenage girl gets tutored by a handsome older male; if that formula bothers you, so will this book.

The author does a good job of covering electricity basics, including the same equation that Rudi van Drunen covered in his opening column, which relates power to current and voltage. But this book goes further into circuits and explains many things in more detail, providing, for example, the equation for resistance, explaining both positive and negative phase shift, and even offering a section on different types of power generation that includes types of batteries.

Fujitaki discusses physics and chemistry where appropriate, keeping things simple, of course. But his explanation of how dopants are used to create N or P type semiconductors actually cleared things up for me. Since he provided lists, such as a ranking of materials that produce static charge, I missed seeing a similar list of elements that are appropriate for use in batteries.

The book includes text sections that review the material covered in cartoons in great depth and actually make the book work. If you want an easy primer on electricity, from circuit breakers to nuclear power generation, this book might be for you.