Book Reviews

ELIZABETH ZWICKY, WITH TREY DARLEY, SAM STOVER, AND PETER BAER GALVIN

Beyond Bullet Points, 3rd Edition

Cliff Atkinson

Microsoft Press, 2011. 240 pp.
ISBN 978-0735620520

I am hoping that, someday, the existence of resources about how to create presentations that are not mind-numbing and the existence of which millions of people and the rising tide of presentation hate (mostly directed at PowerPoint) will have a result. So far, what I have gotten is people who make defensive jokes about "death by PowerPoint" and then put up giant heaps of classic PowerPoint slides—but with clip art of people.

This book is aimed squarely at the worst perpetrators of death by PowerPoint, which is probably good for relieving my future suffering, but does mean that I found it had an offputting marketing flavor at times. Oddly, the fact that it is intentionally PowerPoint-specific, and I prefer Keynote, didn't bother me at all, possibly because it leans strongly on features that Keynote implemented before PowerPoint. Users of older versions of PowerPoint are going to be sadly out of luck, however, since the techniques presented are based around both presenter mode (where you see your notes and the slides, the audience sees just the slides) and multiple masters. Keynote users will have to do some translation and do some things by hand that the author provides scripts for, but it doesn't look horribly onerous.

I probably won't adopt the approach wholesale, but I did pick up some information that will change the way I do slides (possibly with the exception of the cases where I was already being intentionally outrageous). I like the emphasis on story, on speaking casually, on consistency of metaphor, on considering the audience, on not cluttering up your slides. I adore the idea that it discusses actual research, with references and everything, and while it probably oversimplifies the neurology, it's not horrifyingly improbable or irrelevant. (This is not as low a bar as it may seem.) Instead of saying "90% of communication is non-verbal!" it cites particular studies that showed specific improvements for slides with relevant pictures.

It also tries hard to teach people who are used to working with words how to think of images and how to find and choose images. Too many books tell you to use pictures but leave you with no idea where to get the things from, which leads to lame clip-art, copyright violation, and desperation. I'm not sure that the advice on image and on metaphor selection are sufficient for people who don't have experience, but those are difficult issues to teach, and at least *Beyond Bullet Points* tries.

If you are looking to learn how to give presentations well, and you're willing to take it seriously, I recommend that you read this book. Even if you don't end up using this system exactly, you will get a good idea of the important issues. The system described is a lot of work, but it's not make-work; doing presentations well just is a lot of work, no matter how you do it.

Me and My Web Shadow: How to Manage Your Reputation Online

Andrew Mayfield

A&C Black Publishers, 2010. 185 pp.
ISBN 9-781-4081-1908-2

This is a book for people who are not terribly technical on managing their Web presence, from the point of view of somebody who is a real netizen. That's not a word I find myself using often, but I don't know how else to talk about somebody who gets the Internet without being technical. He's a marketing guy who's also capable of non-judgmentally advising that buying posts for your blog is probably a really bad idea and that if you want to be part of a community you are going to have to do real work. (He's a UK marketing guy, and I was taken aback initially by his saying that a good reputation required "graft"; that would be hard work, not bribery.)

Although I'm not the target audience, I found some of the advice useful and the book as a whole reassuring and appropriate. It encourages people to view the Internet as a public space where you want to exercise some care and taste and behave like a responsible, contributing human, and it gives you advice on how to do that (including on how to recover from errors). I would certainly offer this book to friends who were thinking maybe they wanted to get some control over this Internet thing and get some use out of it, and feel confident that it would lead them, gently, towards behavior I consider appropriate.

I have some quibbles, of course. It's a bit Google-centric for my taste (note that I'm employed by Yahoo! and may be more sensitive to this than other people). And while I appreciate the thought, I don't really believe that people should change their passwords every six months; I believe they should make them not completely stupid and different everywhere.

Head First Networking

Al Anderson and Ryan Benedetti

O'Reilly and Associates, 2009. 487 pp.
ISBN 978-0-596-52155-4

On the up side, this is a networking book aimed at people teaching you to do actual TCP/IP networking, instead of teaching you to pass a certification exam. It does a good job of walking you through how routers and switches work, and what subnet masks and routing tables actually do, which are difficult concepts for people. It takes you through real problems that network administrators face, emphasizing troubleshooting and network design, which are the hard parts, and leaving the OSI model, which almost never clarifies anything for anybody, to an appendix.

It has one major down side, which is that it has an amazing number of errors. Most of them are just annoying, but some of them have the potential to actually confuse readers; I would only recommend it to relatively resilient learners. For instance, it has a nice example about tracing an intercepted message back to the computer that originated it, but when you get as far as the router, there is a missing explanation, leading to a baffling moment where the problem is solved by an apparent miracle. And it says that routers protect you from MAC address spoofing, which is not precisely wrong but is totally misleading. If there's a router in the path from client to server, the server won't be fooled by MAC spoofing at the client end, but only because the server cannot pay any attention to the MAC in the first place, not because the router is exerting some protective effect.

There is also a design oddity—*Head First Networking* is electron first networking, with quite a lot of discussion of waveforms on cables and how you turn electrical impulses into ones and zeroes before you ever get to the packets. That's a justifiable decision, and there are certainly people for whom it is the best approach. There are also people who are going to tune out somewhere around the multimeter. That's probably a mistake; try skipping ahead to the packets and see if it picks up for you.

While this is never going to be for everyone, if it were fixed, it would be a great resource for many people. It's a much more realistic introduction to networking than most resources.

Hunting Security Bugs

Tom Gallagher, Bryan Jeffries, and Lawrence Landauer

Microsoft Press, 2006. 592 pp.
ISBN 978-0-7356-2187-9

I feel kind of silly recommending a security book from 2006. Five years is a really long time. Unfortunately, the changes in many of the topics the book covers are minimal. XSS? Check. SQL injection? Check. Buffer overflows? Apparently as resilient as cockroaches. Sadly, people are even still running five-year old browsers. Some things are guaranteed to be timeless; canonicalization is good, blacklists are bad. The net result is that *Hunting Security Bugs* is Microsoft-oriented and leaves out some important attacks (XSRF, clickjacking, Flash and PDF vulnerabilities) but still manages to provide a solid introduction to what people get wrong and how you find it.

If you have a significant technical background and are interested in securing software on Microsoft platforms, this is a good place to look. It will show you how to think like a security tester and introduce relevant tools. It also goes through common counter-arguments ("But nobody will make a hostile server!"). It goes into the innards of most things enough to show you why things are vulnerable. It assumes the reader is capable of reading C++ code and has some basic idea what things are bad (it does not explain terms like "escalation of privilege"), but it does not assume much network background.

Pro Puppet

James Turnbull and Jeffrey McCune

Apress, 2011. 318 pp.
ISBN 978-1-4302-3057-1

Perhaps, like me, you have been a mite stymied by the endless debates between proponents of different configuration management tools over the past decade. It seems that config management is destined to be every bit as partisan as choice of editor. If you've been in the field a while you've doubtless noticed there's been a steady uptick in "ssh and a for loop" jokes in the past couple of years. I sense that we passed an inflection point in the past 18 months or so. (Sure, there's still plenty of room for theorists to debate how many angels can dance on the head of a DSL.) But seeing how young people coming to the field seem to view being a sysadmin as passe and insist they are a breed apart—the devop—it seems that the tools themselves are damn near cooked.

If you've been scratching your head over the Puppet-Chef-CFEngine-Bcfg2-LCFG-etc. question while holding your old shell scripts together with duct tape, ponder no more. Let me tell you, Puppet isn't the only tool but it is a *fine* tool. This book will help you over the initial hurdles. You know the old "give a man a fish, teach a man to fish" chestnut? Some technical books err on the side of all theory, others give you pages of code and cli copy-paste but leave you without understanding what you're doing. The authors of this book struck a nice balance between the two extremes. There's just enough hand-holding to get you going and just enough theory to keep the book around for reference. They walk you through getting your initial management server (aka puppetmaster) up and running, and take you through some real-world scenarios managing various services on several different mainstream platforms. Then they move on to integration with source control, scalability, reporting, integration with third-party tools, and, finally, developing your own modules.

I did find a few gotcha mistakes in my review copy, particularly in the first couple of chapters, which are heavy on the cli copy-pastey bits. Nothing too hard to work around, though. Otherwise, if I had to make one criticism it would be that while there's a sizable base of third-party modules available (via the Puppet Forge Web site), the authors didn't spend much time on how to adapt these modules for your own use. Puppet comes with a good deal already built in, but most people are going to need external modules. The section on using a module from Puppet Forge was a bit weak at three pages; I think it could've been a stand-alone chapter.

To sum up, this is a fine introduction to Puppet. James Turnball's previous book on Puppet, 2008's *Pulling Strings with Puppet,* was badly in need of a rewrite. If you're already a hardcore Puppet user then this book probably won't be very interesting for you. But if you're interested in dropping the duct tape and shell scripts and graduating to a proper configuration management tool, buy this book and give Puppet a try.

—Reviewed by Trey Darley (trey@treyka.net)

Hadoop: The Definitive Guide, 2d Edition

Tom White

O'Reilly Media, Inc., 2010. 626 pp.

ISBN 978-1-4493-8973-4

I've been working with some large-data projects, and one of my co-workers suggested Hadoop. Being new to Hadoop, I jumped on O'Reilly's *Definitive Guide* and never really looked back.

To be honest, I've only read about half of the book so far, but it's become clear that (1) Hadoop is the right solution for my problem and (2) this is the right book for me to use. Chapter 1 walks you through the "hows" and "whys" of Hadoop, which introduces some of the problems of dealing with large data sets, as well as the inception and evolution of Hadoop to address those problems. Chapter 2 introduces you to MapReduce, a "programming model for data processing," which means that MapReduce is the workhorse of Hadoop—it is the mechanism you must write to take raw data and put it into Hadoop. Also in this chapter, a basic comparison between Hadoop and UNIX Tools (AWK) shows the scalability and power that Hadoop allows. In a nutshell, Hadoop handles scalability and redundancy, allowing the user/programmer to focus almost entirely on data issues (indexing, searching, etc.). A big part of the redundancy and scalability comes not from MapReduce, but from the Hadoop Distributed File System (HDFS), which is explained in Chapter 3. HDFS is designed to allow for large file storage (terabytes) and a transparent clustering system, which is the beauty of Hadoop. Increasing storage space simply means adding new systems to the cluster. HDFS separates this from the programmer, once again, allowing them to deal with the data and not worry too much about the underlying infrastructure. Chapter 4 extends this to explaining the mechanics behind Hadoop I/O.

Chapter 5 is a step-by-step walkthrough of building a MapReduce application, which is where everything starts to gel. You begin by building your Map and Reduce functions, and run them against a small subset of your data (as an aside, you do a very similar process in Chapter 2, but with sample data provided by the authors). Once you feel that everything is working as it should, you run your application on a cluster against the entire data set. Hopefully, there's more tuning and less troubleshooting at this point, since it can be difficult to identify bugs when dealing with tons of data across the cluster. Chapter 6 goes even deeper, by explaining how MapReduce works at a very low level. This provides for better tuning and more advanced MapReduce functions. Chapters 7 and 8 continue by explaining "MapReduce Types and Formats" and "MapReduce Features," respectively. I haven't spent much time on these chapters, but just skimming through them I can see that there is a lot to learn—and a pretty big difference between setting up a working Hadoop system vs. a finely tuned (and well-programmed) environment.

Chapters 9 and 10 show you how to actually set up a Hadoop Cluster and administer it. The next five chapters deal with various tools that have evolved to make using Hadoop easier:

Pig, Hive, HBase, ZooKeeper, and Sqoop. I've spent some time with Hive, but haven't yet dug into the others. Chapter 16 outlines seven different case studies, including Last.fm, Facebook, and Rackspace. It is truly amazing the amount of data we live with in today's Internet, and Hadoop is a very powerful, cost-effective (free) and useful tool for dealing with it. There are a couple of other Hadoop books out there, but of the ones I perused, this one seems like the right fit. It's well written, very technical, but not intimidating. If you don't work with Hadoop, you probably will, and this is the book to grab when it happens.

—Reviewed by Sam Stover

Hacking Exposed™ Wireless: Wireless Security Secrets & Solutions, Second Edition

Johnny Cache, Joshua Wright, and Vincent Liu

McGraw-Hill, 2010. 512 pp.
ISBN 978-0-07-166661-9

This book is a solid reference for wireless protocols, mechanisms, tools, and techniques. Some of the notable additions from the first edition include Zigbee (yay!) and new tools. I originally picked this book up to help with some Zigbee work I was doing, but ended up skipping around the whole book. There is a fair bit of basic info that you'll find in any wireless book, such as finding 802.11 networks and WEP cracking, so this can serve as a good introduction for beginners as well. For the OS X crowd, there is a decent amount of effort and time spent on explaining and introducing OS X tools and methods. Linux, of course, is also featured throughout.

Three sections divide the book: Hacking 802.11 Wireless Technology, Hacking 802.11 Clients, and Hacking Additional Wireless Technologies. As I mentioned earlier, the third section was the one that most interested me, and if you want to mess around with BlueTooth, Zigbee, and DECT, this is the go-to book for you. If you already have a solid grasp of other 802.11 technologies, you may feel that it's not worth the cost just for the additional wireless technologies, but I was glad to finally have a resource that gives a real introduction to Zigbee hacking, especially with the introduction of Killer-Bee, which is a "Python-based framework for manipulating Zigbee and IEEE 802.15.4 networks."

OK, enough about Zigbee, let me talk a bit about the rest of the book. As is typical with Hacking Exposed books, there are a ton of example scenarios which deal with realistic scenarios, which should be very helpful to the budding wireless pen-tester. Part 1 should be nothing new to the seasoned wireless expert, but lays a solid groundwork and gives updated information on the techniques and tools used. The step-by-step instructions and explanations should make it easy for just about anyone to follow along and learn by doing, which is my favorite way to do it.

Part 2 was a pretty pleasant surprise for me. There are a number of useful client-side tools out there that I wasn't aware of (IPPON, Ferret, and Hamster) and some old-and-still-good tools like Metasploit and Ettercap. As with Part 1, everything was well explained and easy to follow. I especially liked Chapter 6, which walks you through the entire process in OS X.

I've already waxed enthusiastic about Part 3, so I'll spare you more. Overall, this was a solid book with great examples, good overall 802.11 reference material, and enough new stuff to justify springing for the second edition. In fact, I'm anxiously awaiting the third edition to see what they add to the Zigbee/DECT sections.

—Reviewed by Sam Stover

DTrace: Dynamic Tracing in Oracle Solaris, Mac OS X and FreeBSD (Oracle Solaris Series)

Brendan Gregg and Jim Mauro

Prentice Hall, 2011. 1152 pp.
ISBN 978-0132091510

This is the book you need if you are trying to understand performance, and debug performance problems, on a system that contains DTrace system analytics infrastructure. It also includes useful performance analysis methods, questions, and logical exploration that could help a junior or mid-level systems administrator or programmer learn about performance analysis, but of course the scripts would not be of much use.

This review won't spend space waxing eloquent about DTrace, as that has been done before, many times (including many publications by USENIX, as the Google search "site:www.usenix.org dtrace" reveals). I'll just summarize by saying that DTrace is the most important modern-era computing tool for understanding and debugging system behavior and performance.

DTrace itself is mind-bogglingly complex. It includes a new idea, implemented by kernel structures and the new D language. Before this book there were many sources of DTrace information, including the Solaris manual, tutorials, talks, toolkits, and forums. And before this book were other books, such as *Solaris Performance and Tools: DTrace and MDB Techniques for Solaris 10 and OpenSolaris* by Richard McDougall, Jim Mauro and Brendan Gregg. That book concentrated on Solaris performance while exploring DTrace as one of the useful tools.

This book is what people who are interested in DTrace—and even people who are experienced with DTrace—have been waiting for. Those DTrace knowledge seekers now have, in one volume, information on what DTrace is, how to use DTrace, when to use DTrace, and lots of new, useful, informative scripts that can be typed in (or downloaded from the book's Web site) and executed to analyze a system. It also includes, as needed, information from those other sources, such as scripts from the DTraceToolkit. *DTrace* the book builds up knowledge of DTrace the facility from scratch, and quickly, to the point where a reader is able to write useful DTrace scripts and achieve a deep knowledge of system performance analysis.

Because DTrace was part of OpenSolaris and therefore had its source code released, and because it's so powerful, it has been ported to other operating systems, including FreeBSD and Mac OS X. Unfortunately, those ports are not quite as rich as the OpenSolaris implementation, so some information in the book does not apply to them and some scripts don't work on them. The book does a good job of pointing out these limits. For example, there is no tcp provider in Mac OS X, so the scripts in that section, including "tcpconnect," which shows TCP connections as they occur, will not run on Mac OS.

Rather than reading this book, you could start from the very good manual that comes with Solaris. However, that is daunting, complete and complex, and mostly, cleverly avoids showing "how to use DTrace to do useful stuff." Cleverly, because it's a powerful tool, and the authors don't want to limit the audience to using the tool only in the ways they have thought of. This is much like Sawzall showing exactly how to use all the features of its tool but not showing how to use it to cut a hole in a wall. Perhaps that would keep the users from realizing they could also cut holes in the roof, floor, and so on.

Part 1 provides a succinct summary of the language and the other DTrace components, and the remainder of the book shows how to use DTrace to examine various aspects of user and kernel mode operations, how to solve performance issues, and how to diagnose problems. It explores the included scripts line by line and character by character, teaching by example and stressing the learn-by-doing approach. In fact that's how the authors learned DTrace—in the field and within Sun/Oracle, solving customer's problems and writing scripts to make DTrace even more useful and efficient.

The authors are maintaining a Web site from which the scripts can be downloaded and where other information, such as the errata, is likely to be posted over time, at http://dtracebook.com/index.php/Main_Page. The book is also available electronically via Safari and on the Kindle. Either way makes the text available on a computer, which is great for searching as well as for copy-and-paste actions.

In short, these are the kernel innards and performance analysis details you are looking for. The book is a masterpiece of hands-on system performance analysis methodologies and tools. If you don't have Jim Mauro's cell phone number, this book is the next best thing. (Fair notice, Jim is a friend and it *is* nice to have his cell phone number.)

—Reviewed by Peter Baer Galvin