book reviews

ELIZABETH ZWICKY, WITH TONY DEL PORTO, NICK STOUGHTON, AND SAM STOVER

SECURITY DATA VISUALIZATION

Greg Conti

No Starch Press, 2007. 230 pages.

ISBN 978-1-59327-143-5

VISUALIZING DATA

Ben Fry

O'Reilly, 2007. 382 pages.

ISBN 978-0-59651-455-6

If you are a fan of visualization, or perhaps you're looking for a new hobby and have some numbers you're interested in, buy both these books. If you already have the numbers ready to whip into shape and you have tools you love to do it with, you might skip *Visualizing Data*. If you have no interest in security or networks, or you're just starting from scratch, you could skip *Security Data Visualization*. But probably, you want both of them. You want *Security Data Vsualization* if you are drowning in network-oriented data, even if you don't want to be a fan of visualization.

Visualizing Data is a tour through the process of taking a question, finding the numbers that go with it, and producing an interactive visualization of the answer that you can put on a Web page easily. It uses Processing, which is my current favorite play language, but it also talks in passing about other tools and languages. The techniques are applicable to any language, although if you haven't got a language you're really fluent in for this purpose, I'd recommend going ahead and learning Processing. It's easy to learn, and the ability to publish to the Web without hassle is priceless.

My favorite thing about *Visualizing Data* is that it tackles the whole process in all its blood, guts, and gore. It starts with finding the data and cleaning it up. Many books assume that the data fairy is going to come bring you data, and that it will either be clean, lovely data or you will parse it carefully into clean, lovely data. This book assumes that a significant portion of the data you care about comes from some scuzzy Web page you don't control and that you are going to use exactly the minimum required finesse to tear out the parts you care about. It talks about how to do this, and how to decide what the minimum required finesse would be. (Do you do it by hand? Use a regular expression? Actually bother to parse XML?)

Visualizing Data also shows a couple of cool visualization techniques, but it is primarily a process book; it's designed to take people with a casual interest in visualizing data and walk them through the whole process, end-to-end, from finding the data to ending up with a refined, interactive way of looking at it. It also teaches Processing, on the side.

Security Data Visualization has some discussion of the process, but it's mostly a catalog of ways you haven't thought of to look at network and security data. It's most interesting to people who care about network data (even if they don't care about security), but if you're into data visualization, there's a lot there even if networks and security aren't your area. These are interesting data sets in a lot of ways. Obviously, anything that involves people trying to break into computers has inherent dramatic value, but network security data sets are interesting as a problem, as well. They're big and complicated, and people are intentionally hiding stuff in them. Techniques that work here can be adapted to many other large, complicated, multivariate data sets.

If you do work with security data, particularly network data, there's an in-depth explanation of how to use visualization tools to illuminate characteristics of your data, along with a really great references section to point you to more information. It's not a cookbook, but it's not an area where the recipes have been found yet, either.

HANDBOOK OF NETWORK AND SYSTEM ADMINISTRATION

Jan Bergstra and Mark Burgess, editors

Elsevier, 2007. 997 pages.

ISBN 978-0-444-52198-9

The preface to this book says it is written by researchers, for researchers, educators, and advanced practitioners. In this case, you should take this very seriously. It is intended for academic researchers first and foremost, with educators as a strong secondary audience, and a hope that a few practitioners (people you might consider nonacademic researchers, for instance) might have some interest. It's the sort of book where eigenvectors need no explanation.

That makes it hard to evaluate. When I review, I hold in my head a picture of the sort of people I'm reviewing for. For most of those people, for people who do programming and system administration for a living, this book is not of interest. If you spend most of your time administering computer systems, as opposed to doing research and writing papers, what you need to know here is that the field has reached a point where a major academic publisher will publish a serious academic book, and it looks all impressive and stuff. You don't want to own it, unless you want to bludgeon people with it, metaphorically or literally. Since it weighs probably twice what my laptop does, it makes an excellent weapon, and if you'd like people to believe in the field's seriousness, the sheer density of the text ought to convince them.

Furthermore, this kind of book is not going to be welcoming and engaging to anybody. That's not the reason for its existence, and that's not the style academic researchers call for. You can't fault it for that. You also can't fault an anthology for being uneven, and although you can fault some of the individual chapters for being not particularly well focused, you can't be surprised. If you tell somebody to write down all the interesting theoretical stuff you might want to know about system administration and topic X, most of the time you are going to get something that doesn't have a clear focus or audience. The best authors in *Handbook of Network and System Administration* transcend this, but it's difficult to do.

So I am left with a book that I think is excellent in parts and that I think is a fine example of its type, but that I mostly didn't enjoy and that most readers of this review are not going to want to own. I did enjoy several chapters, including, surprisingly, the one on graph theory and, less surprisingly, Matt Blaze's chapter on security models, and I found several others to be of theoretical interest, not to mention the inherent amusement value in statements such as "The belief that one must control specific files seems to arise from a lack of trust in the software base being managed." There is a reason I do not trust the software base I manage. It is the same reason that psychiatric nurses do not trust patients to behave rationally. I have both practical and theoretical cause to believe that software is in no way trustworthy.

If, by happenstance, you are interested in this book and the data visualization books, read this one first, or let a long time elapse between them. Otherwise, the contrast between the beautiful data visualizations and the charts and graphs in this book will drive you mad. They are not, objectively, horrible charts and graphs. But they are for the most part to proper data visualization as Little League baseball is to the World Series.

LINUX FIREWALLS: ATTACK DETECTION AND RESPONSE WITH IPTABLES, PSAD, AND FWSNORT

Michael Rash

No Starch Press, 2007. 290 pages.

ISBN 978-1-59327-141-1

Here's another stark contrast. This is the exact opposite of a theoretical book. It does mention the relevant concepts, but it's primarily a step-by-step tour of iptables, psad, and fwsnort. If you're building a Linux firewall and want to know what all the bells and whistles are, when you might want to set them off, and how to hook them together, here you go.

[Editor's Note: I also read this book. Mike has written several articles for *;login:* about topics he covers in considerable detail in this book: psad, fwknop, and fwsnort. Running on Linux, psad monitors iptables logfiles and can send email alerts and even add in reactive filters in response to network events. Mike wrote psad as a replacement for the aging portsentry Perl script, used to detect scans. But psad does this and much more. With fwknop you can open ports in your firewall in response to a cryptographic request sent to the firewall. The most ambitious tool of all is fwsnort; it turns a Linux iptables ruleset into an IDS using many Snort rules.

Linux Firewalls is more a book about using these tools than an iptables primer. But if you ever wanted to learn more about iptables' less familiar features, such as pattern matching within application data or logging minutia, this is the book for you.]

THE BOOK OF PF: A NO-NONSENSE GUIDE TO THE OPENBSD FIREWALL

Peter N.M. Hansteen

No Starch Press, 2008. 158 pages.

ISBN 978-1-59327-165-7

REVIEWED BY TONY DEL PORTO

PF is the OpenBSD packet filter. I've used PF for a variety of things since its release with OpenBSD 3.0 and like it for its intuitive syntax and comprehensive feature set. Although the system manual and FAQ are complete in describing PF's features, figuring out how to translate those features into a functioning set of rules has required frequent trips to my favorite search engine. *The Book of PF* aims to provide an overview of the things that can be done with PF, with enough examples to get the reader started. It is explicitly not a how-to book or cookbook, but, rather, a less terse (and more enjoyable to read) explanation of PF's features.

After a bit of overview the author covers PF rule syntax in two short chapters and then goes on to describe some of the more interesting features of the current version of PF. In addition to the standard things a stateful packet filter is expected to do, PF has been adapted to deal with a variety of issues that plague networked systems. PF provides greylisting and tarpitting to deal with spam, packet queueing to deal with oversubscribed bandwidth, and failover via CARP. There is an authentication scheme via ssh for network access, NAT and port redirection, proxies for ftp, tftp, and even the initial connection between hosts via the synproxy facility. PF is available on the various *BSDs and the author notes the differences in feature implementation where appropriate. The author ends the book with monitoring and debugging tools and some suggestions for choosing suitable hardware. He also touches briefly on performance, which has been the subject of a previous *;login:* article ("Linux vs. OpenBSD: A Firewall Performance Test," by Adamo and Tablo, December 2005).

On the whole the book is a great resource and has me eager to rewrite my aging rulesets to take advantage of PF's more recent features. In particular spamd (not spamassassin) has my attention. I will pick a few nits though. I'm a little perplexed by the author's decision to place the chapter on wireless networks early in the book and between chapters on more commonly used features, as well as why the debugging section was presented so late. One of the most frequent challenges I've faced is the question, "Why doesn't this network service work?" The debugging section is key in providing tools to answer that question, at least as far as PF is concerned. I would have preferred the organization to be more bread-and-butter up front and icing later on.

For future editions, I would love to see an appendix with some real-world rulesets demonstrating how the features of PF can be combined to express a network policy. The author provides a copious list of both online and dead-tree resources for finding such examples; however, a book that aims to be "a stand-alone document to enable you to work on your machines with only short forays into man pages and occasional reference to the online and printed resources" should include a few multipage rulesets, especially if the book is to be consulted when a ruleset is broken and the interwebs are not reachable for consultation. At 145 pages, the book has room to grow, and I look forward to reading the next edition.

CROSS-PLATFORM DEVELOPMENT IN C++: BUILDING MAC OS X, LINUX, AND WINDOWS APPLICATIONS

Syd Logan

Addison Wesley, 2007. 576 pages.

ISBN: 978-0-321-51437-0

REVIEWED BY NICK STOUGHTON

If I were ever to write a book, there is a very good chance that I would produce a volume that looks a lot like Syd Logan's *Cross-Platform Development in C++*. There is a fundamental principle that my good friend Stephen Walli puts as, "There's no such thing as a portable application, merely applications that have been ported." Syd starts out in the introduction with the point that even a simple "hello, world" application produces different output on Mac OS X and a Windows platform (that pesky line-ending difference from DOS).

The book goes on to describe a set of "Items"—maxims to help improve code portability. Each item is well expounded, and the text is littered with examples. If it were not for the examples, the book need not have had the "in C++" as part of its title. Indeed, the C++ used is sufficiently close to C in almost every case that I would certainly recommend this book to C developers, and quite possibly to developers who work in any similar computer language.

Syd is not actively involved in the same standards committees as I am, and so he leaves "Use Standards based APIs" to item 16, whereas I would have had this as item 1, or at worst 2. He also isn't as up-to-date with them as he might be. For example, he states that the "GNU C library implements all the functions specified in ISO/IEC 9945-1:1996"; although this is true, it also implements all the functions in the 2001 and 2008 editions, and many of the new functions in the 2008 edition have come from the GNU C library. He uses the old _POSIX_SOURCE feature test macro, which was superseded by the POSIX_C_SOURCE macro in 1996. But the correct points are made, and the overall advice is sound.

Syd also recommends using a platform abstraction library; his background is from Netscape, and not surprisingly he pushes the Netscape Portable Runtime Library (NSPR). There are other such libraries, most notably Boost, and although Boost rates a mention, the relative strengths and weaknesses of the differing approaches are not discussed.

All in all, this is a highly recommended book for anyone involved in software development. Do not be put off by the C++ in the title. Do not expect it to answer every question you've ever had. But do expect to be provoked into writing better, more portable code.

METASPLOIT TOOLKIT FOR PENETRATION TESTING, EXPLOIT DEVELOPMENT, AND VULNERABILITY RESEARCH

David Maynor, K.K. Mookhey, Jacopo Cervini, Fairuzan Roslan, and Kevin Beaver

Syngress, 2007. 352 pages.

ISBN: 978-1-59749-074-0

REVIEWED BY SAM STOVER

Metasploit Toolkit is a funny book—not "ha ha" funny, but oddly put together. Chapter 1 is 63 pages long, Chapter 2 is 11 pages, and Chapter 3 is a mere 6. See what I mean? Funny. OK, now that I've covered the only negative thing I can come up with, let's get to the important bits. To put it as concisely as I possibly can, if you are in any way interested in a book on Metasploit, this is the one to get. That's it: Stop looking around and justfind this book. Don't even bother browsing through it at the bookstore; that would be a waste of time. If I could figure out a way to mainline text, I'd recommend that.

So, why do I like this book so much? Well, for starters, it is one of the first books by Syngress that didn't overwhelm me with spelling and grammar mistakes. This could be due to one of two reasons: Either there weren't any, or I was so engrossed in the material presented by the book that I didn't notice them. I'm betting on the former, but YMMV. However, you say there are other Metasploit books out there. This is true, and I've even reviewed some of them, but the ones I've seen don't deal with Metasploit v3.x. Ah, but you counter that the rewrite in Ruby for v3.x didn't affect the user experience, since it was all under the hood. Wrong and *wrong*. I've been using Metasploit pretty much since it came out, but v3.x finally makes it easy to discover your own vulnerabilities, and this book shows you how. This is a huge step for Metasploit, one that truly allows it to compete with the likes of CANVAS and IMPACT. (What's with the all caps? Are they yelling at us? Should Metasploit be METASPLOIT? Someone tell HD he needs to change the name.)

Setting aside the odd chapter structure, you'll find a large portion of this book (over 100 pages' worth) dedicated to case studies. I've said it before, and I'll say it again: I love case studies. They let me set things up, run through the process, and learn from it. Any book that keeps my fingers on the keyboard as much as it keeps my eyes on the page is a keeper. Following the case studies are three glossaries. If you are a veteran user, there's plenty in the first couple of chapters that you can ignore, such as how to install and use Metasploit, but there's also a fair bit of detail on how the 3.x changes will impact your experience. If you are a Metasploit noob, reading the book from cover to cover will effectively migrate you out of noobdom. Each Metasploit component is addressed in sufficient detail to give either the new or the experienced user what they need—the noob learns how things are, and the vet learns how things have changed.

Overall, Metasploit 3.x is a pretty exciting advance in the exploit/vuln landscape. This book is the perfect guide for getting started and/or learning what is new. I honestly can't recommend it enough: it was written precisely the way I like books to be written: clear, concise, and with lots of examples. If you want to learn anything about Metasploit, I can't think of a better place to start.