book reviews

ELIZABETH ZWICKY WITH SAM STOVER AND RIK FARROW

THE COMPLETE APRIL FOOL'S RFCS

Thomas A. Limoncelli and Peter H. Salus, compilers

Peer-to-Peer Communications, LLC, 2007. 390 pages.

ISBN 978-1-57398-042-5

This is one of those books that don't need a review so much as a description. For instance, at home I have a Richard Scarry alphabet book with flaps. It doesn't need reviewing. You either are a toddler, and understand that this is unbelievably perfect, or you aren't, and you don't. Similarly, what we have here is all the RFCs released for April Fools' Day, up through now. Some of you are giggling pleasantly at the thought, and you will enjoy the book. Some of you are intrigued at the idea, and you will enjoy the book, too. Some of you have no idea what an RFC is, and the odds are pretty good you won't find them funny. (If you have no idea what April Fools' day is, it's a day when people make jokes, traditionally in the form of fake news items. For instance, some day you should check out how Google Maps recommends you drive from, say, Albuquerque, New Mexico, to Toulouse, France. Some maps might recommend just not doing this, because there is an ocean in the way. But last April, Google quietly developed some new suggestions.)

On April 1, the otherwise generally sober IETF issues an RFC that is not completely serious. Some of these are very, very famous—the Avian Transport Protocol, for instance: Tired of good old UTP? ATP will take you further, but there are significant latency and packet loss issues because it uses pigeons. Some of them are less well known, although I have actually cited HTCPCP (the HyperText Coffee Pot Control Protocol) before. And one of them, actually one of the funniest ones, is not a joke at all (and was released in November). However, the need to document the NULL encryption protocol left people with a certain space for hijinks.

Frankly, you know you're a geek when you find the NULL encryption protocol documentation funny. I laughed out loud, and immediately felt a bit guilty. However, I also know that many of the other people who visit my house also find this sort of thing funny (since I have been known to read particularly bad bits of review copies of books at dinner parties), which makes this a perfect coffee table book for my sort of household. Your less technical visitors are going to find it utterly baffling. Whether this is a good thing or a bad thing is really your call.

A+, NETWORK+, SECURITY+ EXAMS IN A NUTSHELL: A DESKTOP QUICK REFERENCE

Pawan K. Bhardwaj

O'Reilly and Associates, 2007. 744 pages.

ISBN 978-0-596-52824-9

HEAD FIRST PMP: A LEARNER'S COMPANION TO PASSING THE PROJECT MANAGEMENT PROFESSIONAL EXAM

Jennifer Greene and Andrew Stellman

O'Reilly and Associates, 2007. 644 pages.

ISBN 978-0-596-10234-0

This is not a full review of the *A+, Network+, Security+ Exams in a Nutshell* book, which I did not read in its entirety. It is included here because it makes an interesting contrast to *Head First PMP*.

These two books are both specifically about exams, but they take extremely different approaches. It starts with the subtitles. *A+, Network+, Security+ Exams in a Nutshell* is in fact just what its subtitle says: a desktop reference, something you pick up to look up a specific fact, and then put down again. *Head First PMP*, by contrast, is companionable. It is meant to be read in its entirety, and to actively assist you in passing the exam.

Both books have sample questions, but *A+* prints the question immediately followed by the answer. It is effectively impossible to use these as review questions, because most readers will have noticed the answer by the time they've finished reading the question. *Head First PMP* prints all the questions relating to a chapter, has a page break, and then shows all the answers, so that it's easy to actually try to answer the questions without interference. In addition, *A+* gives the correct answer and then a brief restatement of why that answer is correct. *Head First PMP* gives the correct answer and then an explanation of how you should have gotten to that answer given the question and the other answers. (It will point out distractor information in the question, show you how you could eliminate other answers, and so on.)

Both books have exercises you can do to prepare for the test. *Head First PMP* includes these exercis-

es—things such as writing new exam questions, doing crossword puzzles, and various sorts of other questions. *A+* suggests that you go out and get test computers and then try things, except when that's not plausible, in which case the author suggests asking your local administrator. Some of the exercises strike me as merely unhelpful with a multiple-choice test, some of them are pointless (you're supposed to pick up a laser-printed page and note that it's warm, for instance), and some of them are likely to try your administrator's patience well beyond breaking point—"Contact the system or network administrator. Determine which networking protocol is used in the network and why."

Both books talk explicitly about the exams. *A+* provides tips direct from the testing organization ("Read the questions slowly and carefully"), whereas *Head First PMP* talks about question design, suggests a good place in the testing procedure to write down the formulas you'll need, and often delves into peculiarities of the test.

Both books have mistakes, too. In *Head First PMP*, I caught several bloopers in test questions—probably two or three places where it would say something like "The right answer is B, because . . ." and give an explanation that matched answer D, instead. This sort of error is fairly benign. You get confused for a moment, but it sorts itself out if the text is good. I'd prefer perfection, but this is well within my expectations for a first printing. I didn't catch any of these errors in *A+*, probably because I didn't read all the questions, but I did catch a couple of content errors. Take the description of Mac OS X:

"The MAC OS X is used primarily on Apple Macintosh computers. Apple has recently released the Intel version of MAC OS X that can be installed in place of Windows on Intel-based personal computers. . . . The applications that run on Apple computers running the MAC OS need to be written specifically for the MAC OS platform. This is due to the fact that the technology behind microprocessors used in Apple computers is entirely different from the technology used in Intel microprocessors."

Confused? Well, even if you know nothing about OS X, you should be, since the passage contradicts itself. It is also factually wrong and conceptually wrong. No, OS X can't be installed in place of Windows; yeah, all right, you can do some fiddling and install an operating system with some code overlap, but don't try to run the OS X installer. And conceptually, it would be mighty handy to actually understand executable compatibility, otherwise known as why you can install Windows and your favorite UNIX-derived operating system on the very same hardware and still not be able to run the same programs on both.

Both books are faced with the peculiarities of the exams they are about. Exams don't match to the real world perfectly, and these exams have some pretty spectacular divergences from day-to-day life. For instance, I work in a small startup. Let's just say that our project lifecycle does not involve 44 well-defined processes. In fact, I have worked as a consultant for a large corporation with PMPs as project managers, and we still didn't use the full process that the PMP exam talks about. The PMP exam also uses some specialized terminology, sometimes using a term more broadly or more narrowly than people in the industry normally do. *Head First PMP* is up front about this. It says straight out that not all organizations use all parts of the process and that you need to learn them all just in case, and it notes when terms are used unexpectedly. The *A+* book takes the exam's point of view and behaves as if the exam maps well to normal usage, which leads to statements such as "A [peer-to-peer] network is also known as a workgroup. . . . These networks are suitable for only about 10 computers. . . . A network operating system (NOS) does not need to be installed on any computer." This is very specialized usage of the terms "peer-to-peer" and "network operating system." Outside the *A+* context, these words refer to very different things, and these sentences are delusional, the kind of thing that make you think not "What a skilled network technician!" but "I wonder what color the sun is on that planet!" It is unkind to learners to let this sort of thing go by without mentioning it, just as it would be unkind to lead them to expect that every organization has a project management plan and a well-defined change management process for every part of it.

Head First PMP will make a picture out of anything. If it's well suited to a picture, there will definitely be one. Ifit can be depicted graphically (it's a process with inputs and outputs, for instance), it will be. If there have been two pages in a row with no pictures, they'll make up an example and illustrate that. *A+* limits itself mostly to the standard icons the Nutshell Guide series uses and screen shots, although it contains pictures when there's really no other way to show something. But it doesn't go at all out of its way, so that topics such as relative sizes of PC motherboards don't get pictures.

I didn't read *A+ etc. Exams* in its entirety because I don't think I could have. I'm not sure anybody

could, except in very small doses. Admittedly it is labeled "A desktop quick reference" and you don't expect reference books to be gripping end-to-end reads, but there aren't a lot of situations where you need a desktop reference guide to an exam. In fact, the only one I can think of is the situation I find myself in, where I sometimes teach course material that isn't aimed at these exams but where it is a selling point to make it useful for these exams. Being able to look up what the exam covers is useful information for me, and the book is perfectly adequate for that.

I read *Head First PMP* in its entirety, and I did the majority of the exercises. I'm pretty certain that if I read it again and did all the exercises, I'd pass the exam. I also learned some useful stuff about project management, as well as about the PMP exam.

I'd recommend *Head First PMP* to people who're interested in the PMP exam. I'd really only recommend the *A+* book to people who really need a desktop reference to the relevant exams.

INSIDE NETWORK PERIMETER SECURITY

Stephen Northcutt, Lenny Zeltser, Scott Winters, Karen Kent, and Ronald W. Ritchey

Sams Publishing, 2005. 660 pages.

ISBN 0-672-32737-6

I was predisposed not to like this book. It has too many authors, to start with, and then I glanced at the title and thought, "That's interesting, a book on security inside the network perimeter," and was disappointed to discover that it was meant to be an inside look at network perimeter security.

But actually, it's a good look at network security (with a concentration on perimeter security). The writing is a little bit uneven, but not terribly, and there are only a few places where the seams between chapters show as information gets repeated with an inconsistent slant. It covers the important stuff, including some often-neglected topics such as logging, troubleshooting, and VPN integration. And it treats current hype with the restraint it deserves, acknowledging that "intrusion prevention" is a good thing when used judiciously but not a staggering work of genius that will save the world.

Its chapter on auditing underestimates both the naiveté of novice testers and the fragility of average networks—yes, it does include "Get consent, because people have been arrested and convicted for testing security without permission," but it fails to say things such as "Most networks include networked devices that are not multi-purpose computers and those devices tend to have extremely fragile IP implementations." Pretty much every network scan I've ever seen has brought something crashing down, with varying degrees of havoc. The document scanner that disabled its Ethernet interface whenever you pinged it so that the most gentle and considerate of network scans caused it to disappear was merely comic; the friend who brought an entire manufacturing line down was considerably more scarred by the process. Oh, and by the way, if you war-dial a large organization, there will be somebody at work and they will notice. The results of this can also range from comic to deeply disturbing, but deeply disturbing is pretty well guaranteed if you're running during the day and lots of people notice. They develop conspiracy theories, and complain a lot, because frankly, it's very annoying to have the phones all ring one after another.

The final chapters are the weakest, ending with an extended riff on the castle metaphor, which is strained to start with and is entirely abandoned at the end.

Overall, this book provides a good overview of network perimeter security. I'd recommend it to a reasonably experienced network administrator who wanted to understand the whole picture of network security.

AMPLIFYING YOUR EFFECTIVENESS: COLLECTED ESSAYS

Gerald M. Weinberg, James Bach, and Naomi Karten, editors

Dorset House, 2000. 134 pages.

ISBN 0-932633-47-1

As you can see from the summary information, this is a small book, and not a recent one. It was also an accident (I don't know whether I slipped or the publisher did, but it's certainly not the book I intended to request). But it was there, and I read things, so I read it. And I had a really good time reading it, with some moments of enlightenment.

For sheer fun, I recommend the project management haiku, some of which manages to be not only funny but actually insightful: "If a project fails/but we keep working on it/has it really failed?"

My favorite for insight is "Good Practice Hunting," which is about why "best practice" is going to depend on what and where you're practicing. I also love the explanation of the Satir change model, showing why it is that group productivity goes down and people get all weird when you replace the awful broken system they're used to with a nonawful system. These are both essays that I can see myself shoving on people who need to understand their concepts.

This books deals mostly with highly accessible thinking about technical management issues; it would be good for technical managers and the people who have to manage them (from above or below).

ENDPOINT SECURITY

Mark S. Kadrich

Addison-Wesley Professional, 2007. 383 pages.

ISBN-10: 0-321-43695-4; ISBN-13: 978-0-321-43695-5

REVIEWED BY SAM STOVER (SAM.STOVER@GMAIL.COM)

I'm going to start this review by being as honest as I possibly can. When I got this book, I was convinced that it was going to be an utter waste of time. Boy, was I wrong. I found this work to be engaging, interesting, informative, provocative, and most of all fun. I don't necessarily agree with everything the author says, but I do think he's asking the right questions. For example, in Chapter 2, page 25, in a section titled "We're Not Asking Vendors Hard Questions," he lists two questions that he always asks when a vendor is peddling a product to him: (1) What type of systems development life cycle (SDLC) do you use? and (2) What software analysis tools do you use to discover coding flaws in your software?

I don't know about you, but I think those are two fine questions, and I've added them to my repertoire. This book hones in on the problem of security in a way I find very interesting. We all know we have a problem, but the people who are selling us the fixes are dependent upon the problem to stay in business. After exploring that topic in the first three chapters, the author moves into a discussion of where the problems really should be solved—at the endpoint (hence the name of the book). Chapters 4–6 deal with accepting that endpoints are the "real" targets, and how to start building an environment where endpoint security is the goal. To be sure, it doesn't make much sense to spend millions of dollars on the latest and greatest network monitoring devices if your endpoints are ignored. I don't think the author is saying that the monitoring devices are unnecessary, just that it is too easy to forget about the endpoint—and that's where the action is.

Chapter 7, "Threat Vectors," briefly discusses applications and operating systems as means to compromise. Security geeks will find this chapter fairly commonplace, but I think the author does a good job of getting the point across that users, and their desire for more/faster/better applications, ultimately drive the insecurity market.

Chapters 8 through 12 address the different kinds of endpoints and provide a laundry list of things to examine for each type. Each chapter is built uniformly, with emphasis on system checks, hardening measures, application gotchas, and more. This is really the meat of the book—the chapters leading up to this really serve to justify the attention that we have to give to our endpoints, and these chapters provide a step in the right direction. The author addresses Windows, OS X, Linux, PDA/SmartPhones, and embedded devices. I found the methods and suggestions to be informative and beneficial, especially for new users or those unfamiliar with a particular OS.

The paradox, however, is that this book starts something that cannot be finished. Operating systems and applications are always going to evolve and change, and we'll constantly be playing catch-up in order to keep our endpoints secure. I'm not disagreeing with the author—I think endpoint security is extremely critical. But user education, which this book also advocates, is another important part of the equation. You can have the best protected system around, but if the user willingly installs unknown software, the gig is up.

In all, this is a well-written book that's relatively short and an easy read. I found the author's sense of humor to be dry and witty. The constant comparisons between security devices and home heating and/or plumbing devices are fitting and humorous. Although I'm not sure that securing the endpoint will end all woes, it is definitely a process that is too easily overlooked in the search for the Ultimate Security Solution and needs more attention than it is getting. This book should definitely serve as a primer for anyone looking to move in that direction.

REVERSING: SECRETS OF REVERSE ENGINEERING

Eldad Eilam

Wiley, 2005. 624 pages.

ISBN 0-7645-7481-7

REVIEWED BY SAM STOVER (SAM.STOVER@GMAIL.COM)

It has been a long time since I have read a book that I like as much as *Reversing*. Too long. But the wait was worth it: This book is amazing. If you are planning on doing any kind of reverse engineering, this is the first book you should buy. It probably won't be the only book, but it should definitely be the first.

The book is divided into four parts: Reversing 101, Applied Reversing, Cracking, and Beyond Disassembly. Each part comprises 2–4 chapters, with 13 chapters in all. Experienced reversers probably

won't have to read the book from start to finish, and honestly, I don't think anyone else will either. It's my experience that people get into reversing because they have a goal in mind, and the sections discussed in this book are diverse enough that you'll probably want to skip right to the part that interests you. Go right ahead—that's exactly what I did. I went right to Chapter 8 (Reversing Malware), read enough to realize that I didn't know enough, and went back to the Reversing 101 chapters and built a foundation.

There are four chapters in the Reversing 101 section. The first provides a short intro that gives an explanation of reversing techniques, as well as a discussion on justification. Reversing has a bad connotation in some circles, so it's a good idea to familiarize yourself with those circles before you start reversing everything and anything you can get your hands on. This section rounds out with an overview of low-level software and Windows OS fundamentals, then a solid chapter on the tools that you'll be using.

For me, the second part was where this book really came into its own. The Applied Reversing chapters are unique in that they each addresses reversing from a different angle. This provides insight into the various reasons *why* people reverse. Chapter 5 emphasizes reversing for product compatibility because it is sometimes impossible to obtain support or documentation for an API with which you need to interface. Reversing that API could be the answer you need to quickly obtain enough information to ensure that your program uses it effectively. The whole chapter is dedicated to reversing a set of undocumented Windows APIs and learning how they work. Chapter 8 approaches reversing from the standpoint of understanding how malware works—and we all know that malware authors have no intention of making our jobs any easier when it comes to figuring out what they are trying to accomplish. Again, the author makes skillful use of an existing binary (Hacarmy.D backdoor) and steps through the reversing techniques. You can download the backdoor file from the book Web site.

The three Cracking chapters reverse (pun intended) the focus of the book. Now you are the author of the code, and you have to stop other people from reversing your work. The author details how (and why) people crack copy protection, then spends an entire chapter on anti-reversing techniques that you can use to make cracking that much harder. The final section was, quite honestly, a bit out of my reach. Chapter 12 deals specifically with reversing .NET and Chapter 13 addresses decompilation. Not being a .NET or a C programmer by trade, I didn't spend much time with these chapters, but if you need something in these areas there is a ton of info there.

Bear in mind several caveats before grabbing this book. First, in the opening section of Chapter 4, the author outlines the two different reversing methodologies: Offline Code Analysis and Live Code Analysis. In Chapter 5, the author openly acknowledges that this book focuses almost exclusively on Offline Code Analysis instead of "running programs in the debugger and stepping through them." This focus largely results from the suitability of the printed media, and although I understand completely, it was a bit of a letdown. Live Code Analysis, coupled with System Monitoring (discussed in Chapter 4), is much easier for the neophyte reverser, but much harder for the author to write about. Second, I'm sure that the people writing the code that you will be reversing will eventually find out about this book as well, so you can expect that anti-reversing techniques will become more abundant and effective. As with any security technology, the leap-frog of offense and defense will always continue. I just hope it results in another book as good as this one.

Overall, the book was extremely well written, with an extraordinary emphasis on walking through examples of each technique. A lot of the tools and procedures used in each chapter were very similar, but with different goals in mind. This is an exemplary book, and I commend the author on making a voodoo subject approachable to anyone, even me (no small feat).

MYTHS OF INNOVATION

Scott Berkun

O'Reilly Media, 2007. 176 pages.

ISBN-10: 0596527055; ISBN-13: 978-0596527051

REVIEWED BY RIK FARROW

This slim volume shatters many myths I had about invention and innovation. Like many people, I harbored the notion of the "Eureka" moment, a high-intensity experience that will lead to a world-changing invention and that will make me wealthy as a by-product. Berkun dispels this notion quickly, in the very first chapter, as he points to innovator after innovator who spent many years working to come up with the great idea, and many more years developing that idea into a successful product. Euripedes, the Greek who gives us the saying, had worked long and hard to solve the problem of de-

termining whether a gift to a king was pure gold or not. The displacement of the water in his washtub gave him the insight he needed, and the rest is, uh, history.

Berkun's intent is not simply to dispel myths, although he does that with clear and insightful writing. He is also out to teach us about how innovation really works. Berkun also explains how to quickly stamp out innovation via criticism and typical management techniques. He points to environments where innovation thrives, as well as explains why great ideas often languish. I'll give you one hint: Just having an excellent notion is not enough. The history of innovation rarely records failures, but it does tell us that those who succeeded overcame hurdles not only of implementation but also of cultural resistance to change.

If you ever had what you thought was a great idea, or work at a company that considers itself to be in the business of innovation, you owe it to yourself to read this book. It is an easy and pleasant read, and it might be just the thing you need to innovate and follow through on your inspiration—or perhaps just decide that you need to work someplace else.

LIVE LINUX CDS

Christopher Negus

Prentice Hall, 2007. 430 pages.

ISBN 0-13-243274-9

REVIEWED BY RIK FARROW

I will start off by confessing that I didn't read this entire book. I have been using Knoppix, a live CD, for many years now as part of my Linux Hands-On security class, and wondered if this book would make the process of remastering Knoppix CDs any easier. I can say that the chapter on remastering Knoppix did indeed make my life simpler. I found the directions clear and accurate. At one point, I thought I had found a mistake in the instructions, but I had merely misread the book. I wanted a feature to be there, the copying of all files and directories in /etc/skel, but the book says that Knoppix only copies the Desktop and .kde directories, and the book was right.

This book comes with directions for copying and remastering many Live Linux CDs. It also comes with a DVD that contains multiple Live Linux images, so you can boot different versions of Live CDs and see how they work pretty much effortlessly. The early chapters (which I did read) about creating CDs, the Linux boot process, and the many types of Linux Live CDs available were clear and easy reading. I can recommend this book to anyone who prefers a set of up-to-date and proofread instructions over what you can find on the Web.