

Book Reviews

MARK LAMOURINE

Sams Teach Yourself Go in 24 Hours

George Ornbo

Pearson Education, 2018, 464 pages

ISBN 978-0-672-33803-8

You might think that *Sams Teach Yourself Go in 24 Hours* expects you to binge learn a programming language overnight. What Ornbo has in mind is much more reasonable and actually quite digestible. The whole line of Sams' Teach Yourself series runs to over 80 titles, so they have some experience with this format, and the polish shows. These are definitely practical learning manuals for professionals.

Ornbo's book is a prime example of the form. There is very little space given for theory or abstract concepts. The longest chapter is 22 pages, with most around 15 pages. All of the source code shown in the book is also available on GitHub (divided by chapter). Nearly every page has a small box labeled "Try it yourself." These boxes explain the examples, provide instructions, and describe the expected results for the reader.

Each chapter concludes with a short Q&A, a "Workshop" section with quiz questions meant to get the reader to think about or review the chapter, and finally three or four suggested exercises. Some of the exercises are coding practice, but others refer the reader to a video (e.g., Rob Pike's introduction to Go [1]) or further reading (e.g., blog posts, module documentation).

The author really means for you to read each chapter in an hour, spend some time working on the examples, and stop. I'm surprised to find I actually like this format. I'm used to skimming and scanning tech books in a sitting or two and only going back when I think, "Oh, I saw how to do that somewhere." By limiting each chapter to a small topic with tight examples, Ornbo creates truly bite-sized pieces. The upfront time limit reminded me to stop and play and digest each bit. I'm familiar with Go, so I didn't stick to it for all three weeks, but I think that a disciplined reader would do well to take it slow and enjoy the easy pace. I found myself thinking through new possibilities and experiments at the end of each session. The "workshop" and reading exercises do a good job of inspiring curiosity and provide the means to follow up.

Another advantage to the compact "24 hours" format is that it gives Ornbo the chance to talk about more than just the language syntax. He finished the pure language instruction halfway through the book, with chapters on goroutines and channels as Chapters 12 and 13. In the latter part of the book, he spends a chapter on debugging, one on packaging, two on building web services, and one on JSON serialization. There are even two "bonus chapters," available only in the eBook forms, that

describe designing a RESTful API and a chat server using the concepts and techniques presented earlier.

Teach Yourself Go won't replace Donovan and Kernigan's *The Go Programming Language* [2], but it makes a great companion for the professional adding a tool to their box.

Crucial Conversations, 2nd ed.

Kerry Patterson, Joseph Grenny, Ron McMillan, and Al Switzler

McGraw-Hill, 2012, 244 pages

ISBN 978-0-07-177530-4

When I saw the cover of *Crucial Conversations*, where it proclaims "3 Million Copies Sold," my expectations were low. When your number-one recommendation is "We Sold a Lot," I'm prepared to be unimpressed. The whole cover screams "Pop Psychology Self-Help!" All of my skeptic buttons were pushed.

Crucial Conversations is indeed a self-help book but perhaps one that many people in tech could use. The major message is "Think before you speak, especially in high-stakes situations," and I know that's something I'm still learning.

In each chapter, the authors address some aspect of how people interact, in small groups or one on one. First, they introduce a scenario and the participants. Each person comes to the conversation with something at stake, but all have different goals, priorities, and points of view.

At the start of each conversation it is clear that things could easily go off the rails without some care. One person is the primary actor, usually the one with the least power and the most to lose. Through the remainder of the chapter the authors talk about the motivations of each person and how they might react to different approaches to the conversation, with a focus on the goals that the primary actor has and the pitfalls to avoid.

The chapters address how to approach a sensitive conversation and keep the goal in mind, while avoiding the minefields of emotion, defensiveness, cheap jabs, and destructive criticism. The goal isn't to win an argument but to have a dialog and come to a consensus.

I've been reading and learning a lot in the last decade about how my perspective and behavior, both in the workplace and at home, influence the impressions and responses of the people around me. Looking back on my career, I wish I'd understood some of these things long ago, and I struggle to act only in a way that will be constructive (something I did not always do).

The outward trappings of *Crucial Conversations* are those of an upscale business-oriented miracle cure for the climbing executive. The writing style has a similar feel throughout, right down

to the subsection at the end of the first chapter entitled “Our Audacious Claim,” in which the authors talk about the ways you can “Improve Your Relationships” and “Improve Your Health” using their methods.

Crucial Conversations is one of several books by these authors. Together they have formed a company that makes its money selling the books and teaching seminars all over the country. The book feels a lot like a promotion for the seminar, and I came to read it after a friend suggested to her whole company that they should invite the authors to present. I’ll find out more after that occurs.

There is, however, a contrast in content to typical diet, exercise, and pop-psych fads. The scenarios presented here are realistic and not too contrived. The authors really make few magic promises in the rest of the book, concentrating more on how each conversation requires listening, analysis, and self-control to reach the best outcome. Much of the messaging is in line with the tenets and goals of modern Agile management styles.

I don’t know if there is a better way to present these ideas, but if readers can set aside their own wariness, they may find something here to think about and use. I certainly would recommend this book to someone who wants to think more about their own approach to communicating in tense situations.

Linux Hardening in Hostile Networks

Kyle Rankin

Pearson Education, 2018, 242 pages

ISBN 978-0-13-417326-9

One of the ideas I learned in the 1990s working as a sysadmin for a large ISP is “always assume your environment is hostile.” It seems a bit quaint now, but we ran exposed server hosts and desktops and had to learn how to probe and harden them. We assumed our networks were unsafe, and our network group assumed the same thing about all of our infrastructure hosts. We weren’t actively malicious to each other, but we were always wary of our assumptions. It actually made for a great dynamic, and we only had one incident I can recall where someone failed to pay attention and was caught out by the other group.

These days most people rely on their firewalls, both software and hardware, and on virus scanners. I’m not as actively paranoid as I once was just because I don’t have the time. But I haven’t stopped being watchful and curious.

When I picked up *Linux Hardening* I was surprised at how slim it is. I was expecting, for example, a series of rather arcane steps to tweak the behavior of the network stack in the kernel. Apparently, that kind of manual is why Rankin decided to write his own book. In his treatment he does touch on all of the expected topics, but he focuses on the most basic and practical steps to take in each area. What Rankin has realized is that, in the com-

mon focus on network boundaries and kernel exploits, a lot has been lost. It’s not easy to find a high-quality guide to good, basic technical hygiene.

In the first major section of the book, Rankin offers the topics you’d expect: workstation and server security, host networking, and firewalls. The next four chapters touch on common services; web, email, DNS, and database. He closes with a chapter on incident response and a couple of appendices that go into some technical details of Tor and TLS/SSL.

Each chapter presents a progression from the most basic considerations through intermediate tasks and concludes with what Rankin believes are steps that can protect against even government-level attacks (with the caveat that today’s advanced attacks are tomorrow’s script-kiddie tools).

Rankin includes a number of techniques that make perfect sense today but have only become reasonable in the last decade or so. Tails is a modern bootable read-only distribution, and Qubes is designed to isolate each application in its own virtual machine. Both allow the user to control when data is written to disk or is shared between applications. In the networking section, Rankin shows how to create a personal VPN using OpenVPN and introduces Tor. The fact that both have become reasonable things for an intermediate level sysadmin to do just warms my geeky sysadmin heart.

Other high points are the inclusion of a treatment of email transport security using SPF, DKIM, and DMARC, and an entire chapter on securing DNS. The DNS chapter starts with preventing your servers from being co-opted into DDoS attacks and ends with the most practical, compact introduction to DNSSEC that I’ve ever read.

Rankin chooses several tools in the workstation section that have many peers. I would have liked to see some space given to some of the alternatives to his selections. That said, I understand the desire to avoid confusing readers with variety that perhaps doesn’t really improve the learning experience.

Linux Hardening isn’t a deep or comprehensive guide to Linux security, and I wish it had more references to external materials, but it is a good, broad survey of the most important topics. The basic and intermediate tasks are probably within reach of junior or intermediate sysadmins, and the advanced ones would make a good challenge. For the advanced sysadmin there are probably a few nuggets here. I’m off to create a VPN for my home network.

References

[1] <https://www.youtube.com/watch?v=rKnDgT73v8s>.

[2] *The Go Programming Language* (Addison-Wesley Professional, 2015), ISBN 978-0-134-19044-0.