Book Reviews

MARK LAMOURINE, PETER GUTMANN, AND RIK FARROW

Thinking Security: Stopping Next Year's Hackers

Steven M. Bellovin
Addison Wesley, 2016, 382 pages
ISBN 978-0-13-427754-7
Reviewed by Mark Lamourine

It turns out that Steve Bellovin and I have a very similar taste for science fiction. It's common for authors to include an epigraph at the beginning of each chapter to provide a hint at the topic or even some humor. Bellovin's choices come from some of my favorite authors: Lewis Carroll, Poul Anderson, E. E. "Doc" Smith, and Larry Niven among others. He uses many quotes from a recent novelist, Charles Stross. Each of these authors writes about a proposed future or alternate world. They also write about the human implications of their new framework. Stross in particular writes about a near future, extrapolating on current trends in technology and the threats they pose as well as the promise. Bellovin wants us to do the same thing.

In *Firewalls and Internet Security*, Bellovin's previous book on firewalls, he and co-authors William Cheswick and Aviel D. Rubin wrote about a specific known threat and a particular technology to address it. In *Thinking Security* he gives the reader a broad survey of how things stand today, the technologies that protect our systems, and how they can fail us, and he offers some hints about how we might plan for tomorrow.

The narrative arc is what you would expect: understand the problem; survey the technology, uncovering strengths and weaknesses; and consider how to use the strengths to mitigate the dangers. In each chapter, he clearly lays out the topic and cites examples of ways in which each technology has failed. At the end of each chapter he gives a brief summary analysis and conclusions. The book is broken into five sections, which broadly are: the problem statement; the list of technologies considered; a survey of human factors; a set of architectural case-studies; and, finally, a discussion and guidelines for understanding, planning, and pitching good security practices to corporate management.

In the preface, Bellovin calls this book a graduate-level introduction to computer systems security. It is aimed at working system administrators and other security-related IT professionals. He assumes a working level of understanding of how computer systems operate and constantly underpins his text with references to articles and papers that illustrate his topic, with the understanding that readers will follow up if they are not already familiar with the material. Maintaining computer security is an active pursuit.

His motivation in writing this book is a recognition that, in large part, the computer industry is spending resources in ways that do not really improve the security of our systems. This misallocation of effort is understandable but not necessary. His book is a call to arms for system administrators to become versed in the ways in which they can be effective and to make a clear case for good security design and practice over bad. He also wants to impress IT managers to understand and to trust the front-line workers and to support them in their efforts to educate those who make the decisions. Good security is all of our responsibility.

Essential Scrum

Kenneth Rubin
Pearson Education, 2013, 452 pages
ISBN 978-0-13-704329-3
Reviewed by Mark Lamourine

One of the major tenets at the root of all Agilish software development processes is to eliminate unnecessary "ceremony." The idea is that many of the meetings and memos that were the backbone of business processes from the '90s and before have lost their meaning and become largely empty rituals. They consume time without actually conveying information or improving coordination.

But, humans being humans, we love our rituals. People find a truly free-form "just do what you need to" process disconcerting or uncomfortable. If there are no prescribed activities, people create them. It's just my opinion, but I think that's one of the reasons for the popularity of the Scrum method.

Scrum is, in my experience, the most well-defined of the Agile process methods.

In *Essential Scrum*, Rubin presents scrum with the same precision and structure that Scrum offers to the family of Agile methods. His book is a proper reference, presenting the reader with the goals, concepts, and the process of Scrum for software development.

Rubin's approach and tone will suit the business reader and coach. There are no whimsically drawn characters and banter-filled dialogs, which are often used to try to soften the process of learning a new software process framework.

After presenting the conflict that most traditional software development processes create between rigid long-term planning and interrupt-driven priorities (which are the reality of modern software development), Rubin introduces the core device of Scrum: the sprint. He devotes an entire chapter to the concept and purpose of the sprint: to break the work being addressed into manageable chunks and set achievable goals and deadlines. He uses that as the center around which the rest of the Scrum concepts are based.

Three detailed sections follow on how to define the work to be done, prioritize it, and then plan for development and delivery over time. First, Rubin introduces user stories and the ideas of backlog, technical debt, cadence, and velocity. He shows how to define, think about, discuss, and finally agree on a plan and a schedule for work that all of the participants can meet.

In the second of these sections, Rubin explores in detail the roles of the people who participate in the development process, identifying them by their interests in the product and their responsibilities.

This is where Rubin circles back and devotes the final five chapters to the sprint process. He guides the reader through the phases, from planning through execution to retrospective. He closes by noting that Scrum, like most of business, is an endless process, but by providing a set of markers in time, it allows the participants to see and recognize their real accomplishments and keep their eye on the the larger goals of the project.

I usually read and review books with an individual reader in mind, but it is really difficult to gauge how effective a book like this would be for an individual. Books on software development processes are about interactions and communications, and these are not going to be immediately applicable for a reader in isolation no matter how motivated he or she is. For a person who is joining a team already using Scrum this book may be some help. Where it will excel is as a manual for a team lead who is familiar with the process but needs a touchstone to stay grounded while coaching others. Read once, return often, make mistakes, and learn.

Kanban in Action

Marcus Hammarberg and Joakim Sunden
Manning Publications, 2014, 330 pages
ISBN 978-1-617291-05-0

Reviewed by Mark Lamourine

I have mixed feelings sometimes about the "in Action" theme of this series of books. I like books that are either reference or tutorial, and sometimes these books try to span the two types without really reaching the goals of either. Kanban, though, fits the "in Action" slot perfectly.

Hammarberg and Sunden begin by introducing the process itself, then laying down the philosophy that guides the process. They don't shy away from discussing the pitfalls and mistakes that new kanban participants can make. Most significantly, they finish with a section on teaching kanban, closing the loop with the reader.

Kanban is a group process. It centers around the kanban board and the cards, which represent tasks, but the heart of kanban is the process and the culture it builds. Kanban requires practice and diligence until the process becomes comfortable and innate. Then it will no longer seem like something imposed, but rather a natural way of thinking when organizing and managing work for a group.

Whenever I can, I like to review both the dead-tree and ebook versions of books. Personally, I like the experience of paper, but I know others who prefer searchable media. Often there are significant differences in the presentation, especially in the graphics and the code sample rendering on tablets and phones. This is a case where both are effective, but the differences remain.

The graphics in the ebook are clean and full color. This is especially important when the discussion is how to use color to convey information on the board. This is a significant loss in the black-and-white paper copy. With its compactness and searchability, the ebook version would be my choice for a coach or group leader learning and teaching kanban. The paper will be on my shelf to scan and lend.

The authors do a good job of presenting the practice and theory of kanban. They address those learning kanban as team members and leaders, and include frequent sidebars to coaches. I do wish they'd included a section devoted to tips and guidelines for coaches. In my experience, coaching can be a full-time job, and it requires a constant awareness of both the topic under discussion and the "meta topic": keeping the discussion moving and on message. Everything I could want is there, I just wish I could find it in one place.

I said I generally like books that are either tutorial or reference. I think *Kanban in Action* actually will serve both purposes whether you're a member of a team or a new team lead or coach. I'm going to keep both the paper book and the ebook handy.

iOS Application Security: The Definitive Guide for Hackers and Developers

David Thiel
No Starch Press, 2015, 296 pages
ISBN: 978-1-59-327601-0

Reviewed by Peter Gutmann

This book begins with a good, solid backgrounder on iOS development, debugging, and testing that covers the first hundred-odd pages, which was useful for me as a non-iOS developer but which is something that I get the feeling the target audience should know already. It's in Part III, which covers the security aspects of the iOS API, that things get interesting.

For example, there have been a number of studies done on Android that revealed the widespread misuse of TLS, something that's typically done in order to make it easy (or at least easier) to use, but which also renders it totally insecure. The book goes to some lengths to tell developers both how to detect signs of this misuse in other apps and libraries and how to avoid doing it themselves, either by ensuring that the certificate checking is done right or, better, by using certificate pinning in which only specific certificates are trusted rather than anything that turns up signed by a commercial CA. The author's background in security research and pen-testing really comes through here in that he's seen the things that can go wrong and makes a point of addressing these specific issues, rather than just paraphrasing the API documentation.

The rest of the book continues in this manner, providing lots of information and advice to augment the standard documentation on various security-relevant areas, including numerous notes on informal workarounds for issues that developers have discovered over time. In that sense it's a bit like an iOS-security-oriented subset of Stack Overflow, providing all sorts of useful advice to developers that isn't covered in standard documentation.

The next section of the book contains a quick overview of non-iOS-specific issues like buffer and integer overflows, XSS, SQL and XML injection, and so on, standard OWASP issues that are also covered extensively elsewhere. While it's good to at least mention these issues here, given what a hugely complex topic this is and how difficult it is to address in the limited space that's available, it would have been useful to refer readers to more comprehensive coverage like the OWASP Top Ten or CERT's secure coding guides, or for non-free sources, something like *The Art of Software Security Assessment*.

Finally, the book concludes with sections on using (and not misusing) Apple-specific mechanisms like the keychain and dealing with privacy issues around user tracking and unique IDs, extending Apple's not-always-up-to-date-or-completely-accurate documentation in order to give developers best-practice advice on how to get things right.

In summary, this is a book that every iOS developer needs to read and then act on. The next time you see an app that leaks private data everywhere, is vulnerable to a whole host of injection attacks, and uses crypto like it's 1995, ask them why they didn't consult this book before shipping.

The Car Hacker's Handbook: A Guide for the Penetration Tester
Craig Smith
No Starch Press, 2016, 278 pages
ISBN: 978-1-59327-703-1
Reviewed by Rik Farrow

Ever since I got to work with Ian Foster and Karl Koscher on their CAN Bus article (and hear their WOOT '15 presentation), I've found myself wanting to know more about car hacking. Foster and Koscher were working at UCSD while the Jeep hackers, Charlie Miller and Chris Valasek, were devoting a significant chunk of their lives to hacking the Jeep. I wondered whether it was possible to get started in this field, or at least to exercise my curiosity about my own car.

Smith's *Handbook* does a very good job of helping you understand your car's (or a target car's) networking and computing environment. Smith starts out with a simplified description of penetration testing, then heads into his area of expertise: car networks. I was surprised (but shouldn't have been) that there are multiple networks in cars, but pleased to learn that the CAN Bus is the most common and certainly the best documented one. And the Linux kernel has had support for devices that interface to the CAN Bus for many years. Smith spends an entire chapter on explaining how to use Linux tools for communicating with a CAN Bus, as well as another chapter about setting up a testbed environment so you can learn more without risking the device you use to commute to work.

Smith is best when he is describing buses and Electronic Control Units (ECUs) but is not strong when it comes to disassembling binaries. He does provide pointers to tools, hints on how to identify the type of CPU, and so on, but I think his strong point is really on the hardware and communication protocol sides of things.

Even if you aren't interested in becoming a car penetration tester, but you do want to know more about the collection of computers you routinely drive, you would do well to buy and read this book.