Book Reviews

ELIZABETH ZWICKY, WITH EVAN TERAN AND RYAN MACARTHUR

The Visible Ops Handbook: Implementing ITIL in Four Practical and Auditable Steps

Kevin Behr, Gene Kim, and George Spafford

IT Process Institute, 2005. 97 pp.
ISBN 78-0975568613

Don't be fooled by the low page count; there's a lot of information stuffed into that space. It suffers somewhat from having been stuffed. Traditional book design takes a lot more paper, what with the bigger letters and the generous empty borders, but it makes a real difference in readability. However, once you get used to reading the small, densely packed type, you will find that it is a believable account of how to get an IT operation under control.

The authors clearly have experience with real organizations that they have had to actually fix, so they recommend practical steps (for instance, grandfathering every piece of software in use onto the approved list, and slowly paring it down as you move forward). It does what it sets out to do, which is give you a working compromise between ITIL and reality.

Although most system administrators should be able to find something useful here, it's most useful to straight-up IT organizations: people who work in companies that make widgets, or sell widgets, or insure widgets. It maps closely to established computer companies, but if you're at a start-up, you'll have to do some translation, and if you're at a university, the translation is going to have to be pretty liberal.

Visible Ops Security: Achieving Common Security and IT Operations Objectives in Four Practical Steps

Gene Kim, Paul Love, and George Spafford

IT Process Institute, 2008. 108 pp.
ISBN 978-0975568620

Security Metrics: Replacing Fear, Uncertainty, and Doubt

Andrew Jaquith

Addison Wesley, 2007. 299 pp.
ISBN 978-0-321-34998-9

Here are two books on the practicalities of implementing computer security in a big network, which is, sadly, a great deal more about committee meetings than it is about bad guys and exciting technology. Since computer security education is a great deal more about the technology, there is a painful gap to be bridged. *Visible Ops Security* does a good job of bridging those gaps on the purely political end; who do you need to talk to and how do you get them to talk to you? *Security Metrics* is aimed at giving you something to say to management that is also useful to you, which implies that it is strongly based in reality and therefore will not leave you with that vaguely slimy feeling you get from simply making up numbers that seem plausible and support your position.

I like them both a lot. *Visible Ops Security*, not surprisingly, has a lot in common with *The Visible Ops Handbook*, including the dense typesetting and the focus on traditional companies, which may be a drawback for people for whom auditors are mythical beings. Still, it's hard to beat something that suggests ways of getting cooperation. Furthermore, it has a good background on what it's like when security relationships aren't working, which you may find therapeutic—it's always nice to know you're not alone.

Security Metrics also has the voice of a real practitioner (genuine sarcasm, frustration, and silliness included) and not only talks about metrics good, bad, and dangerously fictional, but also gives advice on what to do with them once you've collected them. This includes extremely basic statistics and somewhat more advanced charting advice. As puzzling as this may seem if you're fluent in statistics, knowing about medians and quartiles can be life-changing if you've been stuck wondering why the mean is so poor at characterizing any data you care about, and remembering vaguely that there are supposed to be some other averages out there.

Neither book is going to give you a detailed recipe that you can immediately apply to your organization and be better, safer, and happier next week. This is at least in part because no recipe for doing this exists. Each book gives you some guidelines you can use to construct and implement a plan that will make your site safer and happier next year, which is quite audacious and useful enough.

Mining the Talk: Unlocking the Business Value in Unstructured Information

Scott Spangler and Jeffrey T. Kreulen

Pearson Education, 2007. 208 pp.
ISBN 978-0-13-233953-7

Book reviewing is full of surprises. A lot of them are sad, but every so often you find an unexpected pleasure, and this book is one. I was ready for a lot of things when I started reading it, but what I got was utter practicality; here's an algorithm I kind of knew about (clustering) with all the information you need to turn it into an effective tool for slicing and dicing types of data sources that don't lend themselves to exploitation with regular expressions.

You know that request queue that you're sure is trying to tell you something useful, but nobody can quite figure out what? (Somehow, those categories you predefined just aren't working.) These guys know why it doesn't work and what you can do that will work, so that with an entirely reasonable amount of effort you can sort the requests into categories and correlate those categories against whatever else you've got lying around in the request data. Getting from there to something you can fix is your problem.

What you get here is the algorithm and the practical advice on the tools you surround it with to make things work. There's a sample implementation you can download (it's Java, but set up for a PC), and a bunch of descriptions of the kinds of problems you can apply it to, and what happens when you do that. Some of the problem categories were of more interest to me than others, but I'm happy to have a shiny new tool to play with.

Picturing the Uncertain World: How to Understand, Communicate, and Control Uncertainty Through Graphical Display

Howard Wainer

Princeton University Press, 2009. 227 pp.
ISBN 978-0-691-13759-9

I like this book, but not as much as I would have liked the book that I was imagining, based on the title. This is a book about statistics, graphics, and understanding, something that fans of Edward Tufte will like, with a really nice chapter on how to make graphs that accurately depict significant differences and insignificant differences. It has practical moments, but for the most part, it's more about background than it is about advice and techniques. It's an enjoyable wander through statistics-related topics with particular attention to uncertainty, but not dense with useful illumination.

Web Application Obfuscation

Mario Heiderich, Eduardo Alberto Vela Nava, Gareth Hayes, and David Lindsay

Syngress, 2010. 282 pp.
ISBN 978-1597496049

Despite what you may assume from the title, this book is not about obfuscating Web applications themselves, but instead talks about things from the attacker's perspective. In other words, it answers the question, "How can I make my attack code get by the various types of filters put in place by Web applications?" Normally, I am a low level, native code guy, so this was a nice change of pace from my usual reading.

This book is broken down into a few distinct parts:

- Introduction (Chapter 1)
- Language-level attacks (Chapters 2–7)
- Bypassing Web application firewalls (WAFs) and client-side filters (Chapter 8)
- Mitigation techniques (Chapter 9)
- Future developments (Chapter 10)

This breakdown is actually a very nice way to structure the book. If you're looking for some information specific to PHP, for example, then you know exactly which chapter to skip to (Chapter 6). Odds are, you will want to read the whole book, but having the information available in an easy-to-look-up form is nice too.

Chapter 1 is nothing more than a brief introduction to the concepts that will be discussed. The authors review why Web applications need to filter and some of the approaches that developers may use. Since regular expressions are a cornerstone concept for filtering, an overview of how they work is covered as well.

The next few chapters are where the really interesting stuff is. The book does a great job of discussing and comparing the various ways different user agents handle invalid markup and code. Sometimes one stands out as handling a particular situation better than the rest, but inevitably they all fail in some way, allowing the attacker to bypass the filters and still get the user agent to do what they want.

The authors discuss many concepts which, to be honest, I would not have thought of myself. Of particular interest was the concept of making the last character of your injected HTML the first byte of a multibyte unicode codepoint. For several browsers, this caused the next character in the HTML to get "swallowed" up, effectively removing it from the markup! That's pretty cool.

Things get a little crazy (in a good way) once we start into JavaScript and VBScript. The authors slowly ease us into the basics, and then throw us into the deep end with writing JavaScript which uses hexdecimal and octal escapes and, finally, "non-alphanumeric JavaScript," which allows the creation of valid code that is truly incomprehensible to the average person.

In the end, there are two core concepts being expressed in the book. First, for all of the covered languages there are ways to write code which has non-obvious functionality, often in the form of alternate ways of expressing characters (e.g., hexadecimal encoding). Secondly, even though the Web is based on many standards, the implementation of these standards varies, often wildly, when given unusual situations to deal with. This is especially true for HTML and JavaScript.

—Evan Teran

The IDA Pro Book: The Unofficial Guide to the World's Most Popular Disassembler

Chris Eagle

No Starch Press, 2008. 608 pp.
ISBN 978-159327-178-7

At 608 pages, the mere size of this book is intimidating; I would call it a tome. Before picking it up I had heard it was the "IDA Bible," which prompted me to acquire it. My job sometimes involves reverse engineering malware, and so I use IDA. I was a self-starter and never took a training class or watched an online IDA training video. I wish I had picked up this book right when I started reverse engineering, because it not only provides a taxonomy of IDA features but is also a primer on reversing pitfalls and how to overcome them leveraging IDA.

The book is separated into six parts, all of which are extremely valuable. Most technical books I pick up today I read haphazardly, but this book is one of the rare ones I read cover to cover (like Stevens's *TCP/IP Illustrated*). Hard to use at times, but rewarding in the end, this book is undeniably packed with useful examples, interesting asides, expert commentary, and real-world use cases.

Eagle spends no time explaining assembly (if you are using a disassembler, I would hope you are already familiar with whatever instruction set you are working with), and most of the examples are focused on x86-based Portable Executables running on Windows (although there are Linux ELF examples).

The "Advanced IDA Usage" section, one of my favorite parts, walks you through the process of creating an IDS file for OpenSSL, something I've now done, and it has actually improved my reversing capabilities, because statically linked OpenSSL code is now detected automatically. "Real-World Applications," another favorite section, was an amazing surprise, discussing common anti-debugging and anti-static analysis techniques, and how to mitigate these with IDA. All of the examples are available on http://www.idabook.com.

The author even delves into the black art of creating your own processor modules and demonstrates this capability by producing a Python module. Think of a processor module as the implementation of an instruction set architecture in IDA. Understanding how to write a processor module can come in handy when working with malware that uses custom virtualization handlers, which make IDA's x86 processor modules worthless. This topic is tough to find good documentation on; before this book, reading the other processor modules was the only way to learn. Not anymore: crack open this book and you will be on your way to writing your own processor modules, custom loaders, and more.

Every little nook and cranny was accounted for here, and it has transformed the way I use IDA. Now I'm armed with a better understanding of every aspect of IDA and can harness each to improve analysis. The current edition covers IDA up to version 5.3. The current IDA is 6.0, and most of the content is still relevant. The section on console mode is obsolete because OS X and Linux now have QT-based user interfaces (as does Windows); having said that, I must concede that console purists might still exist. There was strong focus on well-known processors and file formats, but this book does not advertise methodologies for reversing random firmware images ripped from bizarre hardware. You will probably come away from *The IDA Pro Book* with a better handle on IDA's internals to aid your reversing effort.

—Ryan MacArthur