book reviews

ELIZABETH ZWICKY, SAM STOVER, AND MING CHOW

THE PRACTICE OF SYSTEM AND NETWORK ADMINISTRATION, 2ND EDITION

Thomas A. Limoncelli, Christine J. Hogan, and Strata R. Chalup

Addison-Wesley, 2007. 938 pages.
ISBN 978-0-321-49266-1

The first edition of this book was also the first book that talked about system administration conceptually, not focusing on theory or nitty-gritty instructions, but delving into information about how system administrators—serious, practicing system administrators, not any old bozo who happened to be saddled with the title—think about the operating system–independent issues involved in the job. It was a ground-breaking book and rightfully generated a lot of excitement, which makes it a hard act to follow.

However, the first edition was by no means a perfect book. The second edition isn't either, but it's a nice improvement. This is the book you need if you're looking for help in being a system administrator: not knowing what command to run, but knowing where to start in dealing with the people and machines around you. I don't agree with everything in it, I would have ordered it differently, and I really wish it had been better edited. But these are all quibbles. The second edition covers more ground, and covers it better, than the first edition.

RELEASE IT! DESIGN AND DEPLOY PRODUCTION-READY SOFTWARE

Michael T. Nygard

Pragmatic Bookshelf, 2007. 333 pages.
ISBN 978-0-9787392-1-8

This is a great book with a bad title and worse back-cover copy. I thought it was going to be about shipping software, a vague concept based on the title and strengthened by a back cover that starts, "Congratulations! Your project is finally finished and ready to ship . . . or is it?" In the terms I'm accustomed to, this book is about building services, not software. Software comes in a box, or on a CD, or you download it; it is shipped. Services you connect to; they are deployed. Yes, the title gets it right, but I was confused, nonetheless, since both get released.

Anyway, this is the book to consult if somebody bops up to you and says, "So, welcome to our new project. In three months, this site is going to go live on the Internet, and when anything goes wrong, your pager is going to go off." In case you don't know, there are two likely outcomes when the site goes live: Either nothing happens, or the site melts down. If nothing happens, either eventually the site melts down, or very, very bad financial things happen and it quietly disappears. You will note that all of these likely outcomes are bad. (They are also not mutually exclusive; probably the most likely outcome involves melting down immediately, melting down again later, and then disappearing eventually.) If you have a pager, your primary goal is to reduce the number of times the site melts down. Your secondary goal is to reduce the amount of time it takes to get the site out of meltdown.

This book will help you with both. It covers the most common mistakes people make in trying to build scalable, reliable sites and describes the ways in which you can add in tolerance, scalability, and manageability. It does so in the unmistakable voice of somebody who has been woken up in the middle of the night to save an ailing computer more times than can be counted.

THE ART OF SOFTWARE SECURITY ASSESSMENT

Mark Dowd, John McDonald, and Justin Schuh

Addison-Wesley, 2006. 1200 pages.
ISBN-10: 0321444426; ISBN-13: 978-0321444424

REVIEWED BY SAM STOVER

In the 1997 movie *Event Horizon*, Laurence Fishburne's character utters the line, "This place is a tomb." I had a very similar tone in my voice as I said, "This book is a tome" when I first laid eyes on *The Art of Software Security Assessment*. This book weighs in at a whopping 1174 pages (including the index); you *know* you're reading a book when you have this juggernaut in your hands. But even more impressive than the size is the content. This book has more value per page than a lot of other vulnerability books I've read, and that is saying something.

The first thing I noticed was that the majority of this book is over my head. Not being well versed in C, I had to struggle through the code to follow what was going on. That said, I found the chapter on Memory Corruption very readable. The chapters on Network Protocols, Firewalls, and Network Application Protocols (14–16, respectively) were especially interesting. Not only do the authors discuss different protocols in-depth, but they also highlight security issues within the protocols. For example, they provide a very complete explanation of how different operating systems treat overlapping IP fragments. This is not a new security flaw, but being walked through the protocol at a low level will definitely give the budding vulnerability researcher something to learn from.

The book is divided into three sections: Introduction to Software Security Assessment, Software Vulnerabilities, and Software Vulnerabilities in Practice. Each section builds on the previous, as the book is designed to be read cover to cover. However, as is always the case, you can feel free to jump around to your areas of need. In the first chapter, you'll read about fundamentals and terms to be used throughout the book. After that, Chapters 2–4 walk through the Design, Operational, and Application Review processes. Once you have a firm grip on how software development flows, the Software Vulnerabilities section jumps into all the problems that can arise during each of the development stages. Tons of sample code and great explanations abound, for both UNIX and Windows environments.

The final section deals with network and application vulnerabilities, and this is where I could really dig in. Understanding how the protocols worked made it a lot easier for me to understand the code samples, but there is still much I don't know, and this book is a great source to learn from. I was continually amazed at how much I learned by reading the individual sections, and yet I'm excited that there is still so much more to discover. I've read and reviewed too many books that left me wanting more. When I walk away from this book wanting more, I'll go tackle something simple like brain surgery.

In summation, I have no complaints at all about this book. Not a single one. I found it challenging, but accessible, and intimidating, but educational. What more could I ask for? I think you'll find that once you tackle this book, many of the other vulnerability development books will fall by the wayside. With this one, you won't need much else—which is good, because if I have to carry any other books besides this tome, I'll have to buy a heftier backpack.

EXPLOITING ONLINE GAMES: CHEATING MASSIVELY DISTRIBUTED SYSTEMS

Greg Hoglund and Gary McGraw

Addison-Wesley, 2007. 384 pp.
ISBN-10: 0132271915; ISBN-13: 978-0132271912

REVIEWED BY MING CHOW

The computer and video gaming industry is a multi-billion-dollar-per-year industry. Since its inception, one problem has plagued both players and the industry: cheating. The spawn of massively multiplayer online role-playing games (MMORPGs) has introduced sophisticated cheating techniques, affecting the fun of games and also the bottom line of game companies. Greg Hoglund and Gary McGraw's latest book, *Exploiting Online Games: Cheating Massively Distributed Systems,* discusses how to cheat and break online games, how to develop some cool gaming hacks, and privacy and legal issues surrounding all the activities.

I was pleasantly surprised at the depth of the book. Many "dark side" topics of game development from modding to building bots were included. Before reading this book, I was only aware of one of the more publicized issues affecting MMORPGs: the sale of virtual goods for real money. There is a chapter in the book dedicated to the issues of money, virtual economies, and criminal activities. There are various technical techniques to exploit a game, including hacking the game's client and user interface, manipulating memory, modifying a game's state, and even using a debugger. The first half of the book (first five chapters) discusses basic issues in online games of which all players and game developers must be aware. The second half of the book (Chapters 6 to 10) discusses hacking various aspects of a game.

The first part of the book provided insightful overviews of (1) why players cheat, (2) relatively low-tech and high-tech ways of cheating in games, and (3) the gaming business and how far companies have gone to prevent piracy of their software. One of the most chilling parts of the book was presented in the second chapter, where the World of Warcraft's Warden, software to combat cheaters but comparable to spyware, was dissected. Hoglund presented a program he wrote called "The Governor" to identify the activities of Warden, including the reading of all open programs, processes, window names, and even memory locations on a player's machine. The book details new ways of cheating in online games, including gold duplication, traveling and respawning via taking advantage of the game client's bugs, aim and combat bots for first-person shooter games, and bots for online poker games. Legal issues with game hacking were also presented, and several popular end user license agreements were examined. To understand the second part of the book, knowledge of C/C++ and assembly language is absolutely necessary. Hoglund and McGraw delved into hacking game clients, reverse engineering, and bot building. Hacking the game client is not just about controlling the game's user interface; it also involves manipulating graphical rendering information and injecting new code into the client via DLL injection. The details on reversing were rather basic, focusing on searching for strings and identifying assembly code patterns. Not everything in the book pertained to attacking online games in the black hat sense: there was also a chapter on what gamers know as modding, creating characters and new maps. The book concluded by stressing the need for security in the design of online games and urging players to pay attention to these issues with online games.

Most of the examples presented in the book were confined to arguably the most popular MMORPG of our time, World of Warcraft. I also found that the book constantly referred to two previous books in the Addison-Wesley Software Security series: *Exploiting Software: How to Break Code* and *Software Security: Building Security In*. You are almost required to read them before delving into this one, considering the number of references to the software security touchpoints, reversing, and debugging, which are covered more deeply in the previous two books. However, if you have read them already, then this is an outstanding book to apply everything learned to a practical topic.

This is a seminal book. Computer and video gaming is an integral part of our socioeconomics and culture, and the industry is booming at an alarming rate. It is important for both gamers and developers to understand that there are serious security, privacy, and legal pitfalls in online games. The problems are very real, and Hoglund and McGraw did a great job conveying the message that online games cannot be deployed or played naively.