;login:

THE MAGAZINE OF USENIX & SAGE

April 2002 • Volume 27 • Number 2

inside:

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

musings

by Rik Farrow

Rik Farrow provides UNIX and Internet security consulting and training. He is the author of *UNIX System Security* and *System Administrator's Guide to System V.*

rik@spirit.com

The LISA conference in San Diego last December had a profound effect on my life. As usual, it was not so much the scheduled activities as it was the little things that happen while you are there. And I am not talking about the food or trips to see the Pacific Ocean either.

I sat in on some parts of the Security track; no surprise there. I listened to Æleen Frisch say that she considered privileges in Windows NT/2K/XP a wonderful security mechanism, along with ACLs, and that these should be copied and used more. Of course, privileges are capabilities and, like capabilities, no less prone to failure and misuse. The Vixie-cron elevation of privilege exploit, which only worked because of a small mistake in the new Linux capabilities kernel feature, is one example. I don't even want to discuss the complexities created with NT's 27 different privileges, with several more in Win2K. Who needs to be an administrator when debug or restore privilege is at hand?

And these technologies are old. It seems that every time I meet Peter Neumann, he tells me about inventing ACLs for Multics back in the sixties (he obviously doesn't remember me). Capabilities were created in the seventies. If these security features were so good, why don't we have secure systems today?

Security Engineering

The answer to this question lies in *Security Engineering*, a book by Ross Anderson. Greg Rose mentioned this book as a "must read," so I ordered it from Amazon and settled down to read it. "Fifty pages a day," I told myself. And so it went for the first 250 pages or so.

Ross Anderson, who now teaches at Cambridge University, has the type of experience required to write a book like this. Perhaps there are other people who have worked in programming, banking, crypto, and medical privacy fields, but few are also as eloquent. And it is the rare technical book that can keep me awake at night (instead of putting me to sleep), but some sections of his book did keep me up. My wife was amazed.

But then, security is my field, so a book like this is much more significant to me than it would be to most people. What might make this book interesting to the larger computer audience is that it approaches security mechanisms in general, rather than focusing on some aspect of computer security, such as authentication or secure programming.

For example, ever wonder how ATMs work and what mechanisms protect both you and the bank that operates the ATM? Anderson goes into great detail explaining not only what gets encrypted but how keys are managed. Anderson also writes about some failures of the ATM system, including the fact that banks have lied about the reliability of the ATM system and how that impacts people.

Discussions of ATMs touch upon physical tamper resistance, actually a critical part of ATMs, because these systems may be (and often are) placed in hostile locations. Anderson explains just how difficult it is to produce tamper-resistant hardware by discussing the many ways that people have used to read keys hidden in smartcards: physical attacks, timing attacks, power monitoring attacks, and protocol attacks. Anderson does a splendid job by putting all this in perspective. While stealing a key from a smartcard might take weeks of work, using costly machinery, a successful attack can pay off when the target is a widely used key, for example in a pay-TV operation.

Anderson's focus might appear to wander, but it always winds up being highly relevant. You might wonder what banking has to do with creating secure systems. Yet the banking industry has developed policies and practices that successfully prevent or detect large-scale fraud, and there is a lot to learn from bankers about protocols, encryption in practice, and intrusion detection. I was amazed not only that this was true but also by some of the low-tech mechanisms that continue to be used.

Anderson also talks about the failures of systems: what goes wrong and why it went wrong. Sometimes, it may be fraud: for example, phone phreaking with blue boxes or a bank employee managing to steal over a hundred thousand dollars from a little old lady (and later 'fessed up). Or it might be a more general example, what Anderson calls architectural errors. For example, if you are using a PC and are going to digitally sign a request to buy his book from Amazon, how do you know that what your signature has authorized is what appears on the screen? Clicking to complete the transaction may have just digitally signed something inserted by a clever virus (instead of your Amazon order). So, instead of getting his book in the mail, you might instead have remortgaged your house to Mafia Real Estate, Inc. While this sounds like it would be easy to clear up, elsewhere in this book Anderson explains that legislation has been proposed that would give a digital signature the same weight as the physical signature that you handwrite in the presence of witnesses.

Chapter 19 is entitled "Protecting E-Commerce Systems." It covers SSL and SET, but it goes much deeper than that. Anderson points out, rightly so, that most credit card fraud does not involve the Internet or even e-commerce. Some does involve systems where credit card information is stored, but this should never be the same system that is running the Web server. Most fraud is done by insiders, a much more difficult problem to solve by technology – but not by policy and practices.

Most credit card fraud does not involve the Internet or even e-commerce.

Sections 6 and 7 of Chapter 19 had me so excited I could hardly sit still. Network economics explains how networked systems (not real networks) work to foster and maintain monopolies, such as Microsoft. I have often tried to understand this, and to explain it, but Anderson does a marvelous job, based in part on an article by Andrew Odlyzko: *http://www.acm.org/networker/issue/9805/ssnet.html*. In brief, Odlyzko posits that both the Internet and Microsoft have prospered because these technologies "offer an irresistible bargain to a crucial constituency; namely developers, while managing to conceal the burden it places on users."

Anderson also briefly explains the real purpose behind Passport, the part of Microsoft's .NET initiative that will handle authentication and credit card payments for participating Web sites. While the ostensible purpose of Passport is single sign-on, the real purpose is to create a web of vendors, all of whom use Microsoft for clearing transactions. Microsoft can then collect a huge amount of data about buying habits and sell it among participating vendors. A great deal for the vendors, not for the users. Note that Microsoft already collects and collates information about any visit to its many Web sites.

Anderson does not reveal the same level of passion that I feel in regard to certain monopolies. And he even, occasionally, writes something that I know is wrong (for example, most UNIX systems today have passwords stored in publicly readable files, or that a Sendmail bug in 2000 permitted reading the passwords from a protected file).

All-in-all, this book is a must read important for anyone interested in security, not just for computers but for the systems that we interact with in everyday life. A must read.

Another LISA Story

The most moving event of LISA did not appear on the earlier schedules. Bill LeFebvre talked about his experience working as a senior sysadmin for Turner Broadcasting on September 11. LeFebvre explained how a pool of large Solaris systems can be "switched" from one domain to another "quickly" to support expected surges in visits to a Web site. For the most part, these switches can be anticipated and planned for. B – but, obviously, not in the case of terrorist attacks, which are not scheduled, and are obviously not planned for.

CNN is not your normal Web site by any means. During the US working day, it sees an average hit rate of 85,000 hits/minute, with peaks up to 300,000 hits/minute. Between 8:45 and 9 a.m. EDT, the number of hits went from 85,000 to 229,000 per minute, and as the hits continued to pour in, the Solaris servers started to melt down. By switching servers (by changing IP addresses used by load balancers and pointing the servers at different back-end file servers), the system recovered and went on to handle a peak estimated at almost 2 million hits/minute and to serve a record number of pages that day. In the afternoon, Turner sysadmins noticed an increase in the number of visits to the Cartoon Network Web servers, so they switched over added capacity to handle this. At this point LeFebvre's voice cracks (as my eyes tear up while writing this), considering the impact of that day's events on children.

On a side note, I do want to mention that the people responsible for September 11 have not been punished. The actual perpetrators died that day, but the people behind the scheme – for example, those providing the money and the training – have still not been identified. The "war" in Afghanistan has more to do with an oil and natural gas pipeline than with terrorism. More people have died as "collateral damage" in Afghanistan by the end of January 2002 than died in the attacks on the US on 9/11.

Finally, and on a brighter note, I got to chair a panel entitled "So You Want to Write a Book." As anyone who has written a book will tell you (well, almost anyone, as I have met people who really don't work very hard at it), book writing is highly overrated. It is more difficult, time consuming, and emotionally draining than most people consider. And books rarely pay off financially.

Having said that, I would like to remind people that I am still the SAGE short topics editor and will be at least until June. I am getting a taste of just how difficult it is to get firefighters, that is, most system administrators, to sit down and complete what is essentially a single chapter of a book. I am continuing the thread of the existing booklet topics but also attempting to expand them by getting some of the knowledge learned by system administrators into print. I am looking for authors (always) but, in particular, for someone who can write about UNIX user account management, from the basics to the many different packages that have been used at large sites, and has published in USENIX or LISA proceedings; and for someone who can write about network management tools; and so on. If this appeals to you, and you imagine you have the time to write a "chapter" in the book of system administration, contact SAGE at *sagebooklets@usenix.org*.