INTERNATIONAL BESTSELLER

Over 150,000 copies sold in hardcover

Secrets & Lies

Digital Security in a Networked World

With new information about post-9/11 security

Bruce Schneier

Secrets and Lies

DIGITAL SECURITY IN A NETWORKED WORLD

Bruce Schneier

Wiley Publishing, Inc.

Publisher: Robert Ipsen
Editor: Carol Long
Managing Editor: Micheline Frederick
Associate New Media Editor: Brian Snapp
Text Design & Composition: North Market Street Graphics

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Schneier, Bruce, 1963–
Secrets and lies : digital security in a networked world / Bruce Schneier.
p. cm.
"Wiley Computer Publishing."
ISBN 0-471-25311-1 (cloth : alk. paper) ISBN 0-471-45380-3 (paper : alk. paper)
1. Computer security. 2. Computer networks—Security measures. I. Title.
QA76.9.A25 S352 2000
005.8—dc21 00-042252

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

Praise for *Secrets and Lies*

"Successful companies embrace risk, and Schneier shows how to bring that thinking to the Internet."

–Mary Meeker, Managing Director and Internet Analyst, Morgan Stanley, Dean Witter

"Bruce shows that concern for security should not rest in the IT department alone, but also in the business office . . . *Secrets and Lies* is the breakthrough text we've been waiting for to tell both sides of the story."

–Steve Hunt, Vice President of Research, Giga Information Group

"Good security is good business. And security is not (just) a technical issue; it's a people issue! Security expert Bruce Schneier tells you why and how. If you want to be successful, you should read this book before the competition does."

–Esther Dyson, Chairman, EDventure Holdings

"Setting himself apart, Schneier navigates rough terrain without being overly technical or sensational—two common pitfalls of writers who take on cybercrime and security. All this helps to explain Schneier's long-standing cult-hero status, even—indeed especially—among his esteemed hacker adversaries."

–*Industry Standard*

"All in all, as a broad and readable security guide, *Secrets and Lies* should be near the top of the IT required-reading list."

–*eWeek*

"*Secrets and Lies* should begin to dispel the fog of deception and special pleading around security, and it's fun."

–*New Scientist*

"This book should be, and can be, read by any business executive, no specialty in security required . . . At Walker Digital, we spent millions of dollars to understand what Bruce Schneier has deftly explained here."

–Jay S. Walker, Founder of Priceline.com

"Just as *Applied Cryptography* was the bible for cryptographers in the 90's, so *Secrets and Lies* will be the official bible for INFOSEC in the new millennium. I didn't think it was possible that a book on business security could make me laugh and smile, but Schneier has made this subject very enjoyable."

–Jim Wallner, National Security Agency

"The news media offer examples of our chronic computer security woes on a near-daily basis, but until now there hasn't been a clear, comprehensive guide that puts the wide range of digital threats in context. The ultimate knowledgeable insider, Schneier not only provides definitions, explanations, stories, and strategies, but a measure of hope that we can get through it all."

–Steven Levy, author of *Hackers* and *Crypto*

"In his newest book, *Secrets and Lies: Digital Security in a Networked World*, Schneier emphasizes the limitations of technology and offers managed security monitoring as the solution of the future."

–*Forbes Magazine*

Crypto-Gram

Written and published by Bruce Schneier.

A free monthly e-mail newsletter that provides news, summaries, analyses, insights, and commentaries on computer and network security.

Written in the same style as this book, Crypto-Gram provides timely punditry on security issues, a list of interesting URLs, straight talk on breaking news, and general clueful commentary. Join the over 100,000 readers who get their security information from Crypto-Gram.

To subscribe, send a blank message to:

crypto-gram-subscribe@counterpane.com

Or visit:

http://www.schneier.com/crypto-gram.html

Back issues of Crypto-Gram are available at http://www.schneier.com

Privacy policy: Bruce Schneier, Counterpane Internet Security, Inc., and Counterpane Labs will not use the Crypto-Gram mailing list for any other purpose than e-mailing Crypto-Gram. We will not use the mailing list for company marketing, nor will we sell the list to any third parties.

Secrets and Lies

To Karen: DMASC

Preface

I have written this book partly to correct a mistake.

Seven years ago I wrote another book: *Applied Cryptography.* In it I described a mathematical utopia: algorithms that would keep your deepest secrets safe for millennia, protocols that could perform the most fantastical electronic interactions—unregulated gambling, undetectable authentication, anonymous cash—safely and securely. In my vision cryptography was the great technological equalizer; anyone with a cheap (and getting cheaper every year) computer could have the same security as the largest government. In the second edition of the same book, written two years later, I went so far as to write: "It is insufficient to protect ourselves with laws; we need to protect ourselves with mathematics."

It's just not true. Cryptography can't do any of that.

It's not that cryptography has gotten weaker since 1994, or that the things I described in that book are no longer true; it's that cryptography doesn't exist in a vacuum.

Cryptography is a branch of mathematics. And like all mathematics, it involves numbers, equations, and logic. Security, palpable security that you or I might find useful in our lives, involves people: things people know, relationships between people, people and how they relate to machines. Digital security involves computers: complex, unstable, buggy computers.

Mathematics is perfect; reality is subjective. Mathematics is defined;

computers are ornery. Mathematics is logical; people are erratic, capricious, and barely comprehensible.

The error of *Applied Cryptography* is that I didn't talk at all about the context. I talked about cryptography as if it were The Answer™. I was pretty naïve.

The result wasn't pretty. Readers believed that cryptography was a kind of magic security dust that they could sprinkle over their software and make it secure. That they could invoke magic spells like "128-bit key" and "public-key infrastructure." A colleague once told me that the world was full of bad security systems designed by people who read *Applied Cryptography.*

Since writing the book, I have made a living as a cryptography consultant: designing and analyzing security systems. To my initial surprise, I found that the weak points had nothing to do with the mathematics. They were in the hardware, the software, the networks, and the people. Beautiful pieces of mathematics were made irrelevant through bad programming, a lousy operating system, or someone's bad password choice. I learned to look beyond the cryptography, at the entire system, to find weaknesses. I started repeating a couple of sentiments you'll find throughout this book: "Security is a chain; it's only as secure as the weakest link." "Security is a process, not a product."

Any real-world system is a complicated series of interconnections. Security must permeate the system: its components and connections. And in this book I argue that modern systems have so many components and connections—some of them not even known by the systems' designers, implementers, or users—that insecurities always remain. No system is perfect; no technology is The Answer™.

This is obvious to anyone involved in real-world security. In the real world, security involves processes. It involves preventative technologies, but also detection and reaction processes, and an entire forensics system to hunt down and prosecute the guilty. Security is not a product; it itself is a process. And if we're ever going to make our digital systems secure, we're going to have to start building processes.

A few years ago I heard a quotation, and I am going to modify it here: If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology.

This book is about those security problems, the limitations of technology, and the solutions.

HOW TO READ THIS BOOK

Read this book in order, from beginning to end.

No, really. Many technical books are meant to skim, bounce around in, and use as a reference. This book isn't. This book has a plot; it tells a story. And like any good story, it makes less sense telling it out of order. The chapters build on each other, and you won't buy the ending if you haven't come along on the journey.

Actually, I want you to read the book through once, and then read it through a second time. This book argues that in order to understand the security of a system, you need to look at the entire system—and not at any particular technologies. Security itself is an interconnected system, and it helps to have cursory knowledge of everything before learning more about anything. But two readings is probably too much to ask; forget I mentioned it.

This book has three parts. Part 1 is "The Landscape," and gives context to the rest of the book: who the attackers are, what they want, and what we need to deal with the threats. Part 2 is "Technologies," basically a bunch of chapters describing different security technologies and their limitations. Part 3 is "Strategies": Given the requirements of the landscape and the limitations of the technologies, what do we do now?

I think digital security is about the coolest thing you can work on today, and this book reflects that feeling. It's serious, but fun, too. Enjoy the read.

Contents

1

Introduction

It's been over three years since the first edition of *Secrets and Lies* was published. Reading through it again after all this time, the most amazing thing is how little things have changed. Today, two years after 9/11 and in the middle of the worst spate of computer worms and viruses the world has ever seen, the book is just as relevant as it was when I wrote it.

The attackers and attacks are the same. The targets and the risks are the same. The security tools to defend ourselves are the same, and they're just as ineffective as they were three years ago. If anything, the problems have gotten worse. It's the hacking tools that are more effective and more efficient. It's the ever-more-virulent worms and viruses that are infecting more computers faster. Fraud is more common. Identity theft is an epidemic. Wholesale information theft—of credit card numbers and worse—is happening more often. Financial losses are on the rise. The only good news is that cyberterrorism, the post-9/11 bugaboo that's scaring far too many people, is no closer to reality than it was three years ago.

The reasons haven't changed. In Chapter 23, I discuss the problems of complexity. Simply put, complexity is the worst enemy of security. As systems get more complex, they necessarily get less secure. Today's computer and network systems are far more complex than they were when I wrote the first edition of this book, and they'll be more complex still in another three years. This means that today's computers and networks are less secure than they were earlier, and they will be even less

secure in the future. Security technologies and products may be improving, but they're not improving quickly enough. We're forced to run the Red Queen's race, where it takes all the running you can do just to stay in one place.

As a result, today computer security is at a crossroads. It's failing, regularly, and with increasingly serious results. CEOs are starting to notice. When they finally get fed up, they'll demand improvements. (Either that or they'll abandon the Internet, but I don't believe that is a likely possibility.) And they'll get the improvements they demand; corporate America can be an enormously powerful motivator once it gets going.

For this reason, I believe computer security will improve eventually. I don't think the improvements will come in the short term, and I think they will be met with considerable resistance. This is because the engine of improvement will be fueled by corporate boardrooms and not computer-science laboratories, and as such won't have anything to do with technology. Real security improvement will only come through liability: holding software manufacturers accountable for the security and, more generally, the quality of their products. This is an enormous change, and one the computer industry is not going to accept without a fight.

But I'm getting ahead of myself here. Let me explain why I think the concept of liability can solve the problem.

It's clear to me that computer security is not a problem that technology can solve. Security solutions have a technological component, but security is fundamentally a people problem. Businesses approach security as they do any other business uncertainty: in terms of risk management. Organizations optimize their activities to minimize their cost–risk product, and understanding those motivations is key to understanding computer security today. It makes no sense to spend more on security than the original cost of the problem, just as it makes no sense to pay liability compensation for damage done when spending money on security is cheaper. Businesses look for financial sweet spots—adequate security for a reasonable cost, for example—and if a security solution doesn't make business sense, a company won't do it.

This way of thinking about security explains some otherwise puzzling security realities. For example, historically most organizations haven't spent a lot of money on network security. Why? Because the

costs have been significant: time, expense, reduced functionality, frustrated end-users. (Increasing security regularly frustrates end-users.) On the other hand, the costs of ignoring security and getting hacked have been, in the scheme of things, relatively small. We in the computer security field like to think they're enormous, but they haven't really affected a company's bottom line. From the CEO's perspective, the risks include the possibility of bad press and angry customers and network downtime—none of which is permanent. And there's some regulatory pressure, from audits or lawsuits, which adds additional costs. The result: a smart organization does what everyone else does, and no more. Things are changing; slowly, but they're changing. The risks are increasing, and as a result spending is increasing.

This same kind of economic reasoning explains why software vendors spend so little effort securing their own products. We in computer security think the vendors are all a bunch of idiots, but they're behaving completely rationally from their own point of view. The costs of adding good security to software products are essentially the same ones incurred in increasing network security—large expenses, reduced functionality, delayed product releases, annoyed users—while the costs of ignoring security are minor: occasional bad press, and maybe some users switching to competitors' products. The financial losses to industry worldwide due to vulnerabilities in the Microsoft Windows operating system are not borne by Microsoft, so Microsoft doesn't have the financial incentive to fix them. If the CEO of a major software company told his board of directors that he would be cutting the company's earnings per share by a third because he was going to really—no more pretending—take security seriously, the board would fire him. If I were on the board, *I* would fire him. Any smart software vendor will talk big about security, but do as little as possible, because that's what makes the most economic sense.

Think about why firewalls succeeded in the marketplace. It's not because they're effective; most firewalls are configured so poorly that they're barely effective, and there are many more effective security products that have never seen widespread deployment (such as e-mail encryption). Firewalls are ubiquitous because corporate auditors started demanding them. This changed the cost equation for businesses. The cost of adding a firewall was expense and user annoyance, but the cost of not having a firewall was failing an audit. And even worse, a company

without a firewall could be accused of not following industry best practices in a lawsuit. The result: everyone has firewalls all over their network, whether they do any actual good or not.

As scientists, we are awash in security technologies. We know how to build much more secure operating systems. We know how to build much more secure access control systems. We know how to build much more secure networks. To be sure, there are still technological problems, and research continues. But in the real world, network security is a business problem. The only way to fix it is to concentrate on the business motivations. We need to change the economic costs and benefits of security. We need to make the organizations in the best position to fix the problem *want* to fix the problem.

To do that, I have a three-step program. None of the steps has anything to do with technology; they all have to do with businesses, economics, and people.

STEP ONE: ENFORCE LIABILITIES

This is essential. Remember that I said the costs of bad security are not borne by the software vendors that produce the bad security. In economics this is known as an externality: a cost of a decision that is borne by people other than those making the decision. Today there are no real consequences for having bad security, or having low-quality software of any kind. Even worse, the marketplace often rewards low quality. More precisely, it rewards additional features and timely release dates, even if they come at the expense of quality. If we expect software vendors to reduce features, lengthen development cycles, and invest in secure software development processes, they must be liable for security vulnerabilities in their products. If we expect CEOs to spend significant resources on their own network security—especially the security of their customers—they must be liable for mishandling their customers' data. Basically, we have to tweak the risk equation so the CEO cares about actually fixing the problem. And putting pressure on his balance sheet is the best way to do that.

This could happen in several different ways. Legislatures could impose liability on the computer industry by forcing software manufacturers to live with the same product liability laws that affect other