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

industries. If software manufacturers produced a defective product, they would be liable for damages. Even without this, courts could start imposing liability-like penalties on software manufacturers and users. This is starting to happen. A U.S. judge forced the Department of Interior to take its network offline, because it couldn't guarantee the safety of American Indian data it was entrusted with. Several cases have resulted in penalties against companies that used customer data in violation of their privacy promises, or collected that data using misrepresentation or fraud. And judges have issued restraining orders against companies with insecure networks that are used as conduits for attacks against others. Alternatively, the industry could get together and define its own liability standards.

Clearly this isn't all or nothing. There are many parties involved in a typical software attack. There's the company that sold the software with the vulnerability in the first place. There's the person who wrote the attack tool. There's the attacker himself, who used the tool to break into a network. There's the owner of the network, who was entrusted with defending that network. One hundred percent of the liability shouldn't fall on the shoulders of the software vendor, just as 100 percent shouldn't fall on the attacker or the network owner. But today 100 percent of the cost falls on the network owner, and that just has to stop.

However it happens, liability changes everything. Currently, there is no reason for a software company not to offer more features, more complexity, more versions. Liability forces software companies to think twice before changing something. Liability forces companies to protect the data they're entrusted with.

STEP TWO: ALLOW PARTIES TO TRANSFER LIABILITIES

This will happen automatically, because CEOs turn to insurance companies to help them manage risk, and liability transfer is what insurance companies do. From the CEO's perspective, insurance turns variable-cost risks into fixed-cost expenses, and CEOs like fixed-cost expenses because they can be budgeted. Once CEOs start caring about security—and it will take liability enforcement to make them really care—they're going to look to the insurance industry to help them out.

Insurance companies are not stupid; they're going to move into cyberinsurance in a big way. And when they do, they're going to drive the computer security industry…just as they drive the security industry in the brick-and-mortar world.

A CEO doesn't buy security for his company's warehouse—strong locks, window bars, or an alarm system—because it makes him feel safe. He buys that security because the insurance rates go down. The same thing will hold true for computer security. Once enough policies are being written, insurance companies will start charging different premiums for different levels of security. Even without legislated liability, the CEO will start noticing how his insurance rates change. And once the CEO starts buying security products based on his insurance premiums, the insurance industry will wield enormous power in the marketplace. They will determine which security products are ubiquitous, and which are ignored. And since the insurance companies pay for the actual losses, they have a great incentive to be rational about risk analysis and the effectiveness of security products. This is different from a bunch of auditors deciding that firewalls are important; these are companies with a financial incentive to get it right. They're not going to be swayed by press releases and PR campaigns; they're going to demand real results.

And software companies will take notice, and will strive to increase the security in the products they sell, in order to make them competitive in this new "cost plus insurance cost" world.

STEP THREE: PROVIDE MECHANISMS TO REDUCE RISK

This will also happen automatically. Once insurance companies start demanding real security in products, it will result in a sea change in the computer industry. Insurance companies will reward companies that provide real security, and punish companies that don't—and this will be entirely market driven. Security will improve because the insurance industry will push for improvements, just as they have in fire safety, electrical safety, automobile safety, bank security, and other industries.

Moreover, insurance companies will want it done in standard models that they can build policies around. A network that changes every month or a product that is updated every few months will be much

harder to insure than a product that never changes. But the computer field naturally changes quickly, and this makes it different, to some extent, from other insurance-driven industries. Insurance companies will look to security processes that they can rely on: processes of secure software development before systems are released, and the processes of protection, detection, and response that I talk about in Chapter 24. And more and more, they're going to look toward outsourced services.

For over four years I have been CTO of a company called Counterpane Internet Security, Inc. We provide outsourced security monitoring for organizations. This isn't just firewall monitoring or IDS monitoring but full network monitoring. We defend our customers from insiders, outside hackers, and the latest worm or virus epidemic in the news. We do it affordably, and we do it well. The goal here isn't 100 percent perfect security, but rather adequate security at a reasonable cost. This is the kind of thing insurance companies love, and something I believe will become as common as fire-suppression systems in the coming years.

The insurance industry prefers security outsourcing, because they can write policies around those services. It's much easier to design insurance around a standard set of security services delivered by an outside vendor than it is to customize a policy for each individual network. Today, network security insurance is a rarity—very few of our customers have such policies—but eventually it will be commonplace. And if an organization has Counterpane—or some other company—monitoring its network, or providing any of a bunch of other outsourced services that will be popping up to satisfy this market need, it'll easily be insurable.

Actually, this isn't a three-step program. It's a one-step program with two inevitable consequences. Enforce liability, and everything else will flow from it. It has to. There's no other alternative.

Much of Internet security is a common: an area used by a community as a whole. Like all commons, keeping it working benefits everyone, but any individual can benefit from exploiting it. (Think of the criminal justice system in the real world.) In our society we protect our commons—environment, working conditions, food and drug practices, streets, accounting practices—by legislating those areas and by making companies liable for taking undue advantage of those commons. This kind of thinking is what gives us bridges that don't collapse, clean air

and water, and sanitary restaurants. We don't live in a "buyer beware" society; we hold companies liable when they take advantage of buyers.

There's no reason to treat software any differently from other products. Today Firestone can produce a tire with a single systemic flaw and they're liable, but Microsoft can produce an operating system with multiple systemic flaws discovered per week and not be liable. Today if a home builder sells you a house with hidden flaws that make it easier for burglars to break in, you can sue the home builder; if a software company sells you a software system with the same problem, you're stuck with the damages. This makes no sense, and it's the primary reason computer security is so bad today. I have a lot of faith in the marketplace and in the ingenuity of people. Give the companies in the best position to fix the problem a financial incentive to fix the problem, and fix it they will.

ADDITIONAL BOOKS

I've written two books since *Secrets and Lies* that may be of interest to readers of this book:

Beyond Fear: Thinking Sensibly About Security in an Uncertain World is a book about security in general. In it I cover the entire spectrum of security, from the personal issues we face at home and in the office to the broad public policies implemented as part of the worldwide war on terrorism. With examples and anecdotes from history, sports, natural science, movies, and the evening news, I explain to a general audience how security really works, and demonstrate how we all can make ourselves safer by thinking of security not in absolutes, but in terms of trade-offs—the inevitable cash outlays, taxes, inconveniences, and diminished freedoms we accept (or have forced on us) in the name of enhanced security. Only after we accept the inevitability of trade-offs and learn to negotiate accordingly will we have a truly realistic sense of how to deal with risks and threats.

http://www.schneier.com/bf.html

Practical Cryptography (written with Niels Ferguson) is about cryptography as it is used in real-world systems: about cryptography as an engineering discipline rather than cryptography as a mathematical sci-

ence. Building real-world cryptographic systems is vastly different from the abstract world depicted in most books on cryptography, which assumes a pure mathematical ideal that magically solves your security problems. Designers and implementers live in a very different world, where nothing is perfect and where experience shows that most cryptographic systems are broken due to problems that have nothing to do with mathematics. This book is about how to apply the cryptographic functions in a real-world setting in such a way that you actually get a secure system.

http://www.schneier.com/book-practical.html

FURTHER READING

There's always more to say about security. Every month there are new ideas, new disasters, and new news stories that completely miss the point. For almost six years now I've written *Crypto-Gram*, a free monthly e-mail newsletter that tries to be a voice of sanity and sense in an industry filled with fear, uncertainty, and doubt. With more than 100,000 readers, *Crypto-Gram* is widely cited as the industry's most influential publication. There's no fluff. There's no advertising. Just honest and impartial summaries, analyses, insights, and commentaries about the security stories in the news.

To subscribe, visit:

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

Or send a blank message to:

crypto-gram-subscribe@chaparraltree.com

You can read back issues on the Web site, too. Some specific articles that may be of interest are:

Risks of cyberterrorism:
http://www.schneier.com/crypto-gram-0306.html#1

Militaries and cyberwar:
http://www.schneier.com/crypto-gram-0301.html#1

The "Security Patch Treadmill":
http://www.schneier.com/crypto-gram-0103.html#1

Full disclosure and security:
http://www.schneier.com/crypto-gram-0111.html#1

How to think about security:
http://www.schneier.com/crypto-gram-0204.html#1

What military history can teach computer security (parts 1 and 2):
http://www.schneier.com/crypto-gram-0104.html#1
http://www.schneier.com/crypto-gram-0105.html#1

Thank you for taking the time to read *Secrets and Lies.* I hope you enjoy it, and I hope you find it useful.

Bruce Schneier
January 2004