1 The Business and Economics of Security

Consolidation: Plague or Progress

Originally published in Information Security, *March 2008*

This essay appeared as the second half of a point/counterpoint with Marcus Ranum.

We know what we don't like about buying consolidated product suites: one great product and a bunch of mediocre ones. And we know what we don't like about buying best-of-breed: multiple vendors, multiple interfaces, and multiple products that don't work well together. The security industry has gone back and forth between the two, as a new generation of IT security professionals rediscovers the downsides of each solution.

The real problem is that neither solution really works, and we continually fool ourselves into believing whatever we don't have is better than what we have at the time. And the real solution is to buy results, not products.

Honestly, no one wants to buy IT security. People want to buy whatever they want—connectivity, a Web presence, email, networked applications, whatever—and they want it to be secure. That they're forced to spend money on IT security is an artifact of the youth of the computer industry. And sooner or later the need to buy security will disappear.

It will disappear because IT vendors are starting to realize they have to provide security as part of whatever they're selling. It will disappear because organizations are starting to buy services instead of products, and demanding security as part of those services. It will disappear because the security industry will disappear as a consumer category, and will instead market to the IT industry.

The critical driver here is outsourcing. Outsourcing is the ultimate consolidator, because the customer no longer cares about the details. If I buy my network services from a large IT infrastructure company, I don't care if it secures things by installing the hot new intrusion prevention systems, by configuring

the routers and servers as to obviate the need for network-based security, or if it uses magic security dust given to it by elven kings. I just want a contract that specifies a level and quality of service, and my vendor can figure it out.

IT is infrastructure. Infrastructure is always outsourced. And the details of how the infrastructure works are left to the companies that provide it.

This is the future of IT, and when that happens we're going to start to see a type of consolidation we haven't seen before. Instead of large security companies gobbling up small security companies, both large and small security companies will be gobbled up by non-security companies. It's already starting to happen. In 2006, IBM bought ISS. The same year BT bought my company, Counterpane, and last year it bought INS. These aren't large security companies buying small security companies; these are non-security companies buying large and small security companies.

If I were Symantec and McAfee, I would be preparing myself for a buyer.

This is good consolidation. Instead of having to choose between a single product suite that isn't very good or a best-of-breed set of products that don't work well together, we can ignore the issue completely. We can just find an infrastructure provider that will figure it out and make it work—who cares how?

Prediction: RSA Conference Will Shrink Like a Punctured Balloon

Originally published in Wired News*, April 17, 2008*

Last week was the RSA Conference, easily the largest information security conference in the world. More than 17,000 people descended on San Francisco's Moscone Center to hear some of the more than 250 talks, attend I-didn't-try-to-count parties, and try to evade over 350 exhibitors vying to sell them stuff.

Talk to the exhibitors, though, and the most common complaint is that the attendees aren't buying.

It's not the quality of the wares. The show floor is filled with new security products, new technologies, and new ideas. Many of these are products that will make the attendees' companies more secure in all sorts of different ways. The problem is that most of the people attending the RSA Conference can't understand what the products do or why they should buy them. So they don't.

I spoke with one person whose trip was paid for by a smallish security firm. He was one of the company's first customers, and the company was proud to parade him in front of the press. I asked him whether he walked through the show floor, looking at the company's competitors to see if there was any benefit to switching.

"I can't figure out what any of those companies do," he replied.

I believe him. The booths are filled with broad product claims, meaningless security platitudes and unintelligible marketing literature. You could walk into a booth, listen to a five-minute sales pitch by a marketing type, and still not know what the company does. Even seasoned security professionals are confused.

Commerce requires a meeting of the minds between buyer and seller, and it's just not happening. The sellers can't explain what they're selling to the buyers, and the buyers don't buy because they don't understand what the sellers are selling. There's a mismatch between the two; they're so far apart that they're barely speaking the same language.

This is a bad thing in the near term—some good companies will go bankrupt and some good security technologies won't get deployed—but it's a good thing in the long run. It demonstrates that the computer industry is maturing: IT is getting complicated and subtle, and users are starting to treat it like infrastructure.

For a while now I have predicted the death of the security industry. Not the death of information security as a vital requirement, of course, but the death of the end-user security industry that gathers at the RSA Conference. When something becomes infrastructure—power, water, cleaning service, tax preparation—customers care less about details and more about results. Technological innovations become something the infrastructure providers pay attention to, and they package it for their customers.

No one wants to buy security. They want to buy something truly useful—database management systems, Web 2.0 collaboration tools, a company-wide network—and they want it to be secure. They don't want to have to become IT security experts. They don't want to have to go to the RSA Conference. This is the future of IT security.

You can see it in the large IT outsourcing contracts that companies are signing—not security outsourcing contracts, but more general IT contracts that include security. You can see it in the current wave of industry consolidation: not large security companies buying small security companies, but non-security companies buying security companies. And you can see it in

the new popularity of software as a service: Customers want solutions; who cares about the details?

Imagine if the inventor of antilock brakes—or any automobile safety or security feature—had to sell them directly to the consumer. It would be an uphill battle convincing the average driver that he needed to buy them; maybe that technology would have succeeded and maybe it wouldn't. But that's not what happens. Antilock brakes, airbags and that annoying sensor that beeps when you're backing up too close to another object are sold to automobile companies, and those companies bundle them together into cars that are sold to consumers. This doesn't mean that automobile safety isn't important, and often these new features are touted by the car manufacturers.

The RSA Conference won't die, of course. Security is too important for that. There will still be new technologies, new products and new startups. But it will become inward-facing, slowly turning into an industry conference. It'll be security companies selling to the companies who sell to corporate and home users—and will no longer be a 17,000-person user conference.

How to Sell Security

Originally published in CIO*, May 26, 2008*

It's a truism in sales that it's easier to sell someone something he wants than a defense against something he wants to avoid. People are reluctant to buy insurance, or home security devices, or computer security anything. It's not they don't ever buy these things, but it's an uphill struggle.

The reason is psychological. And it's the same dynamic when it's a security vendor trying to sell its products or services, a CIO trying to convince senior management to invest in security or a security officer trying to implement a security policy with her company's employees.

It's also true that the better you understand your buyer, the better you can sell.

Why People Are Willing to Take Risks

First, a bit about Prospect Theory, the underlying theory behind the newly popular field of behavioral economics. Prospect Theory was developed by Daniel Kahneman and Amos Tversky in 1979 (Kahneman went on to win a Nobel Prize for this and other similar work) to explain how people make

trade-offs that involve risk. Before this work, economists had a model of "economic man," a rational being who makes trade-offs based on some logical calculation. Kahneman and Tversky showed that real people are far more subtle and ornery.

Here's an experiment that illustrates Prospect Theory. Take a roomful of subjects and divide them into two groups. Ask one group to choose between these two alternatives: a sure gain of $500 and 50 percent chance of gaining $1,000. Ask the other group to choose between these two alternatives: a sure loss of $500 and a 50 percent chance of losing $1,000.

These two trade-offs are very similar, and traditional economics predicts that whether you're contemplating a gain or a loss doesn't make a difference: People make trade-offs based on a straightforward calculation of the relative outcome. Some people prefer sure things and others prefer to take chances. Whether the outcome is a gain or a loss doesn't affect the mathematics and therefore shouldn't affect the results. This is traditional economics, and it's called Utility Theory.

But Kahneman's and Tversky's experiments contradicted Utility Theory. When faced with a gain, about 85 percent of people chose the sure smaller gain over the risky larger gain. But when faced with a loss, about 70 percent chose the risky larger loss over the sure smaller loss.

This experiment, repeated again and again by many researchers, across ages, genders, cultures and even species, rocked economics, yielded the same result. Directly contradicting the traditional idea of "economic man," Prospect Theory recognizes that people have subjective values for gains and losses. We have evolved a cognitive bias: a pair of heuristics. One, a sure gain is better than a chance at a greater gain, or "A bird in the hand is worth two in the bush." And two, a sure loss is worse than a chance at a greater loss, or "Run away and live to fight another day." Of course, these are not rigid rules. Only a fool would take a sure $100 over a 50 percent chance at $1,000,000. But all things being equal, we tend to be risk-averse when it comes to gains and risk-seeking when it comes to losses.

This cognitive bias is so powerful that it can lead to logically inconsistent results. Google the "Asian Disease Experiment" for an almost surreal example. Describing the same policy choice in different ways—either as "200 lives saved out of 600" or "400 lives lost out of 600"—yields wildly different risk reactions.

Evolutionarily, the bias makes sense. It's a better survival strategy to accept small gains rather than risk them for larger ones, and to risk larger losses rather than accept smaller losses. Lions, for example, chase young or wounded

wildebeests because the investment needed to kill them is lower. Mature and healthy prey would probably be more nutritious, but there's a risk of missing lunch entirely if it gets away. And a small meal will tide the lion over until another day. Getting through today is more important than the possibility of having food tomorrow. Similarly, it is better to risk a larger loss than to accept a smaller loss. Because animals tend to live on the razor's edge between starvation and reproduction, any loss of food—whether small or large—can be equally bad. Because both can result in death, and the best option is to risk everything for the chance at no loss at all.

How to Sell Security

How does Prospect Theory explain the difficulty of selling the prevention of a security breach? It's a choice between a small sure loss—the cost of the security product—and a large risky loss: for example, the results of an attack on one's network. Of course there's a lot more to the sale. The buyer has to be convinced that the product works, and he has to understand the threats against him and the risk that something bad will happen. But all things being equal, buyers would rather take the chance that the attack won't happen than suffer the sure loss that comes from purchasing the security product.

Security sellers know this, even if they don't understand why, and are continually trying to frame their products in positive results. That's why you see slogans with the basic message, "We take care of security so you can focus on your business," or carefully crafted ROI models that demonstrate how profitable a security purchase can be. But these never seem to work. Security is fundamentally a negative sell.

One solution is to stoke fear. Fear is a primal emotion, far older than our ability to calculate trade-offs. And when people are truly scared, they're willing to do almost anything to make that feeling go away; lots of other psychological research supports that. Any burglar alarm salesman will tell you that people buy only after they've been robbed, or after one of their neighbors has been robbed. And the fears stoked by 9/11, and the politics surrounding 9/11, have fueled an entire industry devoted to counterterrorism. When emotion takes over like that, people are much less likely to think rationally.

Though effective, fear mongering is not very ethical. The better solution is not to sell security directly, but to include it as part of a more general product or service. Your car comes with safety and security features built in; they're not sold separately. Same with your house. And it should be the same with computers and networks. Vendors need to build security into the products

and services that customers actually want. CIOs should include security as an integral part of everything they budget for. Security shouldn't be a separate policy for employees to follow but part of overall IT policy.

Security is inherently about avoiding a negative, so you can never ignore the cognitive bias embedded so deeply in the human brain. But if you understand it, you have a better chance of overcoming it.

Why Do We Accept Signatures by Fax?

Originally published in Wired News*, May 29, 2008*

Aren't fax signatures the weirdest thing? It's trivial to cut and paste—with real scissors and glue—anyone's signature onto a document so that it'll look real when faxed. There is so little security in fax signatures that it's mind-boggling that anyone accepts them.

Yet people do, all the time. I've signed book contracts, credit card authorizations, nondisclosure agreements and all sorts of financial documents—all by fax. I even have a scanned file of my signature on my computer, so I can virtually cut and paste it into documents and fax them directly from my computer without ever having to print them out. What in the world is going on here?

And, more importantly, why are fax signatures still being used after years of experience? Why aren't there many stories of signatures forged through the use of fax machines?

The answer comes from looking at fax signatures not as an isolated security measure, but in the context of the larger system. Fax signatures work because signed faxes exist within a broader communications context.

In a 2003 paper, *Economics, Psychology, and Sociology of Security*, professor Andrew Odlyzko looks at fax signatures and concludes:

> *Although fax signatures have become widespread, their usage is restricted. They are not used for final contracts of substantial value, such as home purchases. That means that the insecurity of fax communications is not easy to exploit for large gain. Additional protection against abuse of fax insecurity is provided by the context in which faxes are used. There are records of phone calls that carry the faxes, paper trails inside enterprises and so on. Furthermore, unexpected large financial transfers trigger scrutiny. As a result, successful frauds are not easy to carry out by purely technical means.*

He's right. Thinking back, there really aren't ways in which a criminal could use a forged document sent by fax to defraud me. I suppose an unscrupulous consulting client could forge my signature on a non-disclosure agreement and then sue me, but that hardly seems worth the effort. And if my broker received a fax document from me authorizing a money transfer to a Nigerian bank account, he would certainly call me before completing it.

Credit card signatures aren't verified in person, either—and I can already buy things over the phone with a credit card—so there are no new risks there, and Visa knows how to monitor transactions for fraud. Lots of companies accept purchase orders via fax, even for large amounts of stuff, but there's a physical audit trail, and the goods are shipped to a physical address—probably one the seller has shipped to before. Signatures are kind of a business lubricant: mostly, they help move things along smoothly.

Except when they don't.

On October 30, 2004, Tristian Wilson was released from a Memphis jail on the authority of a forged fax message. It wasn't even a particularly good forgery. It wasn't on the standard letterhead of the West Memphis Police Department. The name of the policeman who signed the fax was misspelled. And the time stamp on the top of the fax clearly showed that it was sent from a local McDonald's.

The success of this hack has nothing to do with the fact that it was sent over by fax. It worked because the jail had lousy verification procedures. They didn't notice any discrepancies in the fax. They didn't notice the phone number from which the fax was sent. They didn't call and verify that it was official. The jail was accustomed to getting release orders via fax, and just acted on this one without thinking. Would it have been any different had the forged release form been sent by mail or courier?

Yes, fax signatures always exist in context, but sometimes they are the linchpin within that context. If you can mimic enough of the context, or if those on the receiving end become complacent, you can get away with mischief.

Arguably, this is part of the security process. Signatures themselves are poorly defined. Sometimes a document is valid even if not signed: A person with both hands in a cast can still buy a house. Sometimes a document is invalid even if signed: The signer might be drunk, or have a gun pointed at his head. Or he might be a minor. Sometimes a valid signature isn't enough; in the United States there is an entire infrastructure of "notary publics" who officially witness signed documents. When I started filing my tax returns electronically, I had to sign a document stating that I wouldn't be signing my income tax documents. And banks don't even bother verifying signatures on

checks less than $30,000; it's cheaper to deal with fraud after the fact than prevent it.

Over the course of centuries, business and legal systems have slowly sorted out what types of additional controls are required around signatures, and in which circumstances.

Those same systems will be able to sort out fax signatures, too, but it'll be slow. And that's where there will be potential problems. Already fax is a declining technology. In a few years it'll be largely obsolete, replaced by PDFs sent over e-mail and other forms of electronic documentation. In the past, we've had time to figure out how to deal with new technologies. Now, by the time we institutionalize these measures, the technologies are likely to be obsolete.

What that means is people are likely to treat fax signatures—or whatever replaces them—exactly the same way as paper signatures. And sometimes that assumption will get them into trouble.

But it won't cause social havoc. Wilson's story is remarkable mostly because it's so exceptional. And even he was rearrested at his home less than a week later. Fax signatures may be new, but fake signatures have always been a possibility. Our legal and business systems need to deal with the underlying problem—false authentication—rather than focus on the technology of the moment. Systems need to defend themselves against the possibility of fake signatures, regardless of how they arrive.

The Pros and Cons of LifeLock

Originally published in Wired News*, June 12, 2008*

LifeLock, one of the companies that offers identity-theft protection in the United States, has been taking quite a beating recently. They're being sued by credit bureaus, competitors and lawyers in several states that are launching class action lawsuits. And the stories in the media. . . it's like a piranha feeding frenzy.

There are also a lot of errors and misconceptions. With its aggressive advertising campaign and a CEO who publishes his Social Security number and dares people to steal his identity—Todd Davis, 457-55-5462—LifeLock is a company that's easy to hate. But the company's story has some interesting security lessons, and it's worth understanding in some detail.

In December 2003, as part of the Fair and Accurate Credit Transactions Act, or FACTA, credit bureaus were forced to allow you to put a fraud alert on their credit reports, requiring lenders to verify your identity before issuing a credit

card in your name. This alert is temporary, and expires after 90 days. Several companies have sprung up—LifeLock, Debix, LoudSiren, TrustedID—that automatically renew these alerts and effectively make them permanent.

This service pisses off the credit bureaus and their financial customers. The reason lenders don't routinely verify your identity before issuing you credit is that it takes time, costs money and is one more hurdle between you and another credit card. (Buy, buy, buy—it's the American way.) So in the eyes of credit bureaus, LifeLock's customers are inferior goods; selling their data isn't as valuable. LifeLock also opts its customers out of pre-approved credit card offers, further making them less valuable in the eyes of credit bureaus.

And, so began a smear campaign on the part of the credit bureaus. You can read their points of view in this *New York Times* article, written by a reporter who didn't do much more than regurgitate their talking points. And the class action lawsuits have piled on, accusing LifeLock of deceptive business practices, fraudulent advertising and so on. The biggest smear is that LifeLock didn't even protect Todd Davis, and that his identity was allegedly stolen.

It wasn't. Someone in Texas used Davis's SSN to get a $500 advance against his paycheck. It worked because the loan operation didn't check with any of the credit bureaus before approving the loan—perfectly reasonable for an amount this small. The payday-loan operation called Davis to collect, and LifeLock cleared up the problem. His credit report remains spotless.

The Experian credit bureau's lawsuit basically claims that fraud alerts are only for people who have been victims of identity theft. This seems spurious; the text of the law states that anyone "who asserts a good faith suspicion that the consumer has been or is about to become a victim of fraud or related crime" can request a fraud alert. It seems to me that includes anybody who has ever received one of those notices about their financial details being lost or stolen, which is everybody.

As to deceptive business practices and fraudulent advertising—those just seem like class action lawyers piling on. LifeLock's aggressive fear-based marketing doesn't seem any worse than a lot of other similar advertising campaigns. My guess is that the class action lawsuits won't go anywhere.

In reality, forcing lenders to verify identity before issuing credit is exactly the sort of thing we need to do to fight identity theft. Basically, there are two ways to deal with identity theft: Make personal information harder to steal, and make stolen personal information harder to use. We all know the former doesn't work, so that leaves the latter. If Congress wanted to solve the problem

for real, one of the things it would do is make fraud alerts permanent for everybody. But the credit industry's lobbyists would never allow that.

LifeLock does a bunch of other clever things. They monitor the national address database, and alert you if your address changes. They look for your credit and debit card numbers on hacker and criminal websites and such, and assist you in getting a new number if they see it. They have a million-dollar service guarantee—for complicated legal reasons, they can't call it insurance—to help you recover if your identity is ever stolen.

But even with all of this, I am not a LifeLock customer. At $120 a year, it's just not worth it. You wouldn't know it from the press attention, but dealing with identity theft has become easier and more routine. Sure, it's a pervasive problem. The Federal Trade Commission reported that 8.3 million Americans were identity-theft victims in 2005. But that includes things like someone stealing your credit card and using it, something that rarely costs you any money and that LifeLock doesn't protect against. New account fraud is much less common, affecting 1.8 million Americans per year, or 0.8 percent of the adult population. The FTC hasn't published detailed numbers for 2006 or 2007, but the rate seems to be declining.

New card fraud is also not very damaging. The median amount of fraud the thief commits is $1,350, but you're not liable for that. Some spectacularly horrible identity-theft stories notwithstanding, the financial industry is pretty good at quickly cleaning up the mess. The victim's median out-of-pocket cost for new account fraud is only $40, plus ten hours of grief to clean up the problem. Even assuming your time is worth $100 an hour, LifeLock isn't worth more than $8 a year.

And it's hard to get any data on how effective LifeLock really is. They've been in business three years and have about a million customers, but most of them have joined up in the last year. They've paid out on their service guarantee 113 times, but a lot of those were for things that happened before their customers became customers. (It was easier to pay than argue, I assume.) But they don't know how often the fraud alerts actually catch an identity thief in the act. My guess is that it's less than the 0.8 percent fraud rate above.

LifeLock's business model is based more on the fear of identity theft than the actual risk.

It's pretty ironic of the credit bureaus to attack LifeLock on its marketing practices, since they know all about profiting from the fear of identity theft. FACTA also forced the credit bureaus to give Americans a free credit report

once a year upon request. Through deceptive marketing techniques, they've turned this requirement into a multimillion-dollar business.

Get LifeLock if you want, or one of its competitors if you prefer. But remember that you can do most of what these companies do yourself. You can put a fraud alert on your own account, but you have to remember to renew it every three months. You can also put a credit freeze on your account, which is more work for the average consumer but more effective if you're a privacy wonk—and the rules differ by state. And maybe someday Congress will do the right thing and put LifeLock out of business by forcing lenders to verify identity every time they issue credit in someone's name.

The Problem Is Information Insecurity

Originally published in Security Watch, *August 10, 2008*

Information insecurity is costing us billions. We pay for it in theft: information theft, financial theft. We pay for it in productivity loss, both when networks stop working and in the dozens of minor security inconveniences we all have to endure. We pay for it when we have to buy security products and services to reduce those other two losses. We pay for security, year after year.

The problem is that all the money we spend isn't fixing the problem. We're paying, but we still end up with insecurities.

The problem is insecure software. It's bad design, poorly implemented features, inadequate testing and security vulnerabilities from software bugs. The money we spend on security is to deal with the effects of insecure software.

And that's the problem. We're not paying to improve the security of the underlying software. We're paying to deal with the problem rather than to fix it.

The only way to fix this problem is for vendors to fix their software, and they won't do it until it's in their financial best interests to do so.

Today, the costs of insecure software aren't borne by the vendors that produce the software. In economics, this is known as an externality, the cost of a decision that's borne by people other than those making the decision.

There are no real consequences to the vendors for having bad security or low-quality software. 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, it needs to be in their financial best interests to do so. If we expect corporations to spend significant resources on their own network security—especially the security of their customers—it also needs to be in their financial best interests.

Liability law is a way to make it in those organizations' best interests. Raising the risk of liability raises the costs of doing it wrong and therefore increases the amount of money a CEO is willing to spend to do it right. Security is risk management; liability fiddles with the risk equation.

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.

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% shouldn't fall on the attacker or the network owner. But today, 100% of the cost falls directly on the network owner, and that just has to stop.

We will always pay for security. If software vendors have liability costs, they'll pass those on to us. It might not be cheaper than what we're paying today. But as long as we're going to pay, we might as well pay to fix the problem. Forcing the software vendor to pay to fix the problem and then pass those costs on to us means that the problem might actually get fixed.

Liability changes everything. Currently, there is no reason for a software company not to offer feature after feature after feature. Liability forces software companies to think twice before changing something. Liability forces companies to protect the data they're entrusted with. Liability means that those in the best position to fix the problem are actually responsible for the problem.

Information security isn't a technological problem. It's an economics problem. And the way to improve information technology is to fix the economics problem. Do that, and everything else will follow.

Security ROI: Fact or Fiction?

Originally published in CSO Magazine*, September 2, 2008*

Return on investment, or ROI, is a big deal in business. Any business venture needs to demonstrate a positive return on investment, and a good one at that, in order to be viable.

It's become a big deal in IT security, too. Many corporate customers are demanding ROI models to demonstrate that a particular security investment pays off. And in response, vendors are providing ROI models that demonstrate how their particular security solution provides the best return on investment.

It's a good idea in theory, but it's mostly bunk in practice.

Before I get into the details, there's one point I have to make. "ROI" as used in a security context is inaccurate. Security is not an investment that provides a return, like a new factory or a financial instrument. It's an expense that, hopefully, pays for itself in cost savings. Security is about loss prevention, not about earnings. The term just doesn't make sense in this context.

But as anyone who has lived through a company's vicious end-of-year budget-slashing exercises knows, when you're trying to make your numbers, cutting costs is the same as increasing revenues. So while security can't produce ROI, loss prevention most certainly affects a company's bottom line.

And a company should implement only security countermeasures that affect its bottom line positively. It shouldn't spend more on a security problem than the problem is worth. Conversely, it shouldn't ignore problems that are costing it money when there are cheaper mitigation alternatives. A smart company needs to approach security as it would any other business decision: costs versus benefits.

The classic methodology is called annualized loss expectancy (ALE), and it's straightforward. Calculate the cost of a security incident in both tangibles like time and money, and intangibles like reputation and competitive advantage. Multiply that by the chance the incident will occur in a year. That tells you how much you should spend to mitigate the risk. So, for example, if your store has a 10 percent chance of getting robbed and the cost of being robbed is $10,000, then you should spend $1,000 a year on security. Spend more than that, and you're wasting money. Spend less than that, and you're also wasting money.

Of course, that $1,000 has to reduce the chance of being robbed to zero in order to be cost-effective. If a security measure cuts the chance of robbery by 40 percent—to 6 percent a year—then you should spend no more than

$400 on it. If another security measure reduces it by 80 percent, it's worth $800. And if two security measures both reduce the chance of being robbed by 50 percent and one costs $300 and the other $700, the first one is worth it and the second isn't.

The Data Imperative

The key to making this work is good data; the term of art is "actuarial tail." If you're doing an ALE analysis of a security camera at a convenience store, you need to know the crime rate in the store's neighborhood and maybe have some idea of how much cameras improve the odds of convincing criminals to rob another store instead. You need to know how much a robbery costs: in merchandise, in time and annoyance, in lost sales due to spooked patrons, in employee morale. You need to know how much not having the cameras costs in terms of employee morale; maybe you're having trouble hiring salespeople to work the night shift. With all that data, you can figure out if the cost of the camera is cheaper than the loss of revenue if you close the store at night—assuming that the closed store won't get robbed as well. And then you can decide whether to install one.

Cybersecurity is considerably harder, because there just isn't enough good data. There aren't good crime rates for cyberspace, and we have a lot less data about how individual security countermeasures—or specific configurations of countermeasures—mitigate those risks. We don't even have data on incident costs.

One problem is that the threat moves too quickly. The characteristics of the things we're trying to prevent change so quickly that we can't accumulate data fast enough. By the time we get some data, there's a new threat model for which we don't have enough data. So we can't create ALE models.

But there's another problem, and it's that the math quickly falls apart when it comes to rare and expensive events. Imagine you calculate the cost—reputational costs, loss of customers, etc.—of having your company's name in the newspaper after an embarrassing cybersecurity event to be $20 million. Also assume that the odds are 1 in 10,000 of that happening in any one year. ALE says you should spend no more than $2,000 mitigating that risk.

So far, so good. But maybe your CFO thinks an incident would cost only $10 million. You can't argue, since we're just estimating. But he just cut your security budget in half. A vendor trying to sell you a product finds a Web analysis claiming that the odds of this happening are actually 1 in 1,000.

Accept this new number, and suddenly a product costing 10 times as much is still a good investment.

It gets worse when you deal with even more rare and expensive events. Imagine you're in charge of terrorism mitigation at a chlorine plant. What's the cost to your company, in money and reputation, of a large and very deadly explosion? $100 million? $1 billion? $10 billion? And the odds: 1 in a hundred thousand, 1 in a million, 1 in 10 million? Depending on how you answer those two questions—and any answer is really just a guess—you can justify spending anywhere from $10 to $100,000 annually to mitigate that risk.

Or take another example: airport security. Assume that all the new airport security measures increase the waiting time at airports by—and I'm making this up—30 minutes per passenger. There were 760 million passenger boardings in the United States in 2007. This means that the extra waiting time at airports has cost us a collective 43,000 years of extra waiting time. Assume a 70-year life expectancy, and the increased waiting time has "killed" 620 people per year—930 if you calculate the numbers based on 16 hours of awake time per day. So the question is: If we did away with increased airport security, would the result be more people dead from terrorism or fewer?

Caveat Emptor

This kind of thing is why most ROI models you get from security vendors are nonsense. Of course their model demonstrates that their product or service makes financial sense: They've jiggered the numbers so that they do.

This doesn't mean that ALE is useless, but it does mean you should 1) mistrust any analyses that come from people with an agenda and 2) use any results as a general guideline only. So when you get an ROI model from your vendor, take its framework and plug in your own numbers. Don't even show the vendor your improvements; it won't consider any changes that make its product or service less cost-effective to be an "improvement." And use those results as a general guide, along with risk management and compliance analyses, when you're deciding what security products and services to buy.

Social Networking Risks

Originally published in Information Security, *February 2009*

This essay appeared as the first half of a point-counterpoint with Marcus Ranum.

Are employees blogging corporate secrets? It's not an unreasonable fear, actually. People have always talked about work to their friends. It's human nature for people to talk about what's going on in their lives, and work is a lot of most people's lives. Historically, organizations generally didn't care very much. The conversations were intimate and ephemeral, so the risk was small. Unless you worked for the military with actual national secrets, no one worried about it very much.

What has changed is the nature of how we interact with our friends. We talk about our lives on our blogs, on social networking sites such as Facebook and Twitter, and on message boards pertaining to the work we're doing. What was once intimate and ephemeral is now available to the whole world, indexed by Google, and archived for posterity. A good open-source intelligence gatherer can learn a lot about what a company is doing by monitoring its employees' online activities. It's no wonder some organizations are nervous.

So yes, organizations should be concerned about employees leaking corporate secrets on social networking sites. And, as much as I hate to admit it, disciplinary action against employees who reveal too much in public is probably in order. But actually policing employees is almost certainly more expensive and more trouble than it's worth. And when an organization catches an employee being a bit too chatty about work details, it should be as forgiving as possible.

That's because this sort of openness is the future of work, and the organizations that get used to it or—even better—embrace it, are going to do better in the long run than organizations that futilely try to fight it.

The Internet is the greatest generation gap since rock and roll, and what we're seeing here is one particular skirmish across that gap. The younger generation, used to spending a lot of its life in public, clashes with an older generation in charge of a corporate culture that presumes a greater degree of discretion and greater level of control.

There are two things that are always true about generation gaps. The first is that the elder generation is always right about the problems that will result from whatever new/different/bad thing the younger generation is doing. And the second is that the younger generation is always right that whatever they're doing will become the new normal. These things have to be true; the older generation understands the problems better, but they're the ones who fade away and die.

Living an increasingly public life on social networking sites is the new normal. More corporate—and government—transparency is becoming the new normal. CEOs who blog aren't yet the new normal, but will be eventually. And then what will corporate secrecy look like? Organizations will still have secrets, of course, but they will be more public and more open about what they're doing and what they're thinking of doing. It'll be different than it is now, but it most likely won't be any worse.

Today isn't that day yet, which is why it's still proper for organizations to worry about loose fingers uploading corporate secrets. But the sooner an organization can adapt to this new normal and figure out how to be successful within it, the better it will survive these transitions. In the near term, it will be more likely to attract the next-generation talent it needs to figure out how to thrive. In the long term. . .well, we don't know what it will mean yet.

Same with blocking those sites; yes, they're enormous time-wasters. But if an organization has a problem with employee productivity, they're not going to solve it by censoring Internet access. Focus on the actual problem, and don't waste time on the particulars of how the problem manifests itself.

Do You Know Where Your Data Are?

Originally published in the Wall Street Journal, *April 28, 2009*

Do you know what your data did last night? Almost none of the more than 27 million people who took the RealAge quiz realized that their personal health data was being used by drug companies to develop targeted e-mail marketing campaigns.

There's a basic consumer protection principle at work here, and it's the concept of "unfair and deceptive" trade practices. Basically, a company shouldn't be able to say one thing and do another: sell used goods as new, lie on ingredients lists, advertise prices that aren't generally available, claim features that don't exist, and so on.

Buried in RealAge's 2,400-word privacy policy is this disclosure: "If you elect to say yes to becoming a free RealAge Member, we will periodically send you free newsletters and e-mails that directly promote the use of our site(s) or the purchase of our products or services and may contain, in whole or in part, advertisements for third parties which relate to marketed products of selected RealAge partners."

They maintain that when you join the website, you consent to receiving pharmaceutical company spam. But since that isn't spelled out, it's not really informed consent. That's deceptive.

Cloud computing is another technology where users entrust their data to service providers. Salesforce.com, Gmail, and Google Docs are examples; your data isn't on your computer—it's out in the "cloud" somewhere—and you access it from your web browser. Cloud computing has significant benefits for customers and huge profit potential for providers. It's one of the fastest growing IT market segments—69% of Americans now use some sort of cloud computing services—but the business is rife with shady, if not outright deceptive, advertising.

Take Google, for example. Last month, the Electronic Privacy Information Center (I'm on its board of directors) filed a complaint with the Federal Trade Commission concerning Google's cloud computing services. On its website, Google repeatedly assures customers that their data is secure and private, while published vulnerabilities demonstrate that it is not. Google's not foolish, though; its Terms of Service explicitly disavow any warranty or any liability for harm that might result from Google's negligence, recklessness, malevolent intent, or even purposeful disregard of existing legal obligations to protect the privacy and security of user data. EPIC claims that's deceptive.

Facebook isn't much better. Its plainly written (and not legally binding) Statement of Principles contains an admirable set of goals, but its denser and more legalistic Statement of Rights and Responsibilities undermines a lot of it. One research group who studies these documents called it "democracy theater": Facebook wants the appearance of involving users in governance, without the messiness of actually having to do so. Deceptive.

These issues are not identical. RealAge is hiding what it does with your data. Google is trying to both assure you that your data is safe and duck any responsibility when it's not. Facebook wants to market a democracy but run a dictatorship. But they all involve trying to deceive the customer.

Cloud computing services like Google Docs, and social networking sites like RealAge and Facebook, bring with them significant privacy and security risks over and above traditional computing models. Unlike data on my own computer, which I can protect to whatever level I believe prudent, I have no control over any of these sites, nor any real knowledge of how these companies protect my privacy and security. I have to trust them.

This may be fine—the advantages might very well outweigh the risks—but users often can't weigh the trade-offs because these companies are going out of their way to hide the risks.

Of course, companies don't want people to make informed decisions about where to leave their personal data. RealAge wouldn't get 27 million members if its webpage clearly stated "you are signing up to receive e-mails containing advertising from pharmaceutical companies," and Google Docs wouldn't get five million users if its webpage said "We'll take some steps to protect your privacy, but you can't blame us if something goes wrong."

And of course, trust isn't black and white. If, for example, Amazon tried to use customer credit card info to buy itself office supplies, we'd all agree that that was wrong. If it used customer names to solicit new business from their friends, most of us would consider this wrong. When it uses buying history to try to sell customers new books, many of us appreciate the targeted marketing. Similarly, no one expects Google's security to be perfect. But if it didn't fix known vulnerabilities, most of us would consider that a problem.

This is why understanding is so important. For markets to work, consumers need to be able to make informed buying decisions. They need to understand both the costs and benefits of the products and services they buy. Allowing sellers to manipulate the market by outright lying, or even by hiding vital information, about their products breaks capitalism—and that's why the government has to step in to ensure markets work smoothly.

Last month, Mary K. Engle, Acting Deputy Director of the FTC's Bureau of Consumer Protection said: "a company's marketing materials must be consistent with the nature of the product being offered. It's not enough to disclose the information only in a fine print of a lengthy online user agreement." She was speaking about Digital Rights Management and, specifically, an incident where Sony used a music copy protection scheme without disclosing that it secretly installed software on customers' computers. DRM is different from cloud computing or even online surveys and quizzes, but the principle is the same.

Engle again: "if your advertising giveth and your EULA [license agreement] taketh away don't be surprised if the FTC comes calling." That's the right response from government.

Be Careful When You Come to Put Your Trust in the Clouds

Originally published in the Guardian, *June 4, 2009*

This year's overhyped IT concept is cloud computing. Also called software as a service (Saas), cloud computing is when you run software over the Internet and access it via a browser. The salesforce.com customer management software is an example of this. So is Google Docs. If you believe the hype, cloud computing is the future.

But, hype aside, cloud computing is nothing new. It's the modern version of the timesharing model from the 1960s, which was eventually killed by the rise of the personal computer. It's what Hotmail and Gmail have been doing all these years, and it's social networking sites, remote backup companies, and remote email filtering companies such as MessageLabs. Any IT outsourcing—network infrastructure, security monitoring, remote hosting—is a form of cloud computing.

The old timesharing model arose because computers were expensive and hard to maintain. Modern computers and networks are drastically cheaper, but they're still hard to maintain. As networks have become faster, it is again easier to have someone else do the hard work. Computing has become more of a utility; users are more concerned with results than technical details, so the tech fades into the background.

But what about security? Isn't it more dangerous to have your email on Hotmail's servers, your spreadsheets on Google's, your personal conversations on Facebook's, and your company's sales prospects on salesforce.com's? Well, yes and no.

IT security is about trust. You have to trust your CPU manufacturer, your hardware, operating system and software vendors—and your ISP. Any one of these can undermine your security: crash your systems, corrupt data, allow an attacker to get access to systems. We've spent decades dealing with worms and rootkits that target software vulnerabilities. We've worried about infected chips. But in the end, we have no choice but to blindly trust the security of the IT providers we use.

Saas moves the trust boundary out one step further—you now have to also trust your software service vendors—but it doesn't fundamentally change anything. It's just another vendor we need to trust.

There is one critical difference. When a computer is within your network, you can protect it with other security systems such as firewalls and IDSs. You can build a resilient system that works even if those vendors you have to trust may not be as trustworthy as you like. With any outsourcing model, whether it be cloud computing or something else, you can't. You have to trust your outsourcer completely. You not only have to trust the outsourcer's security, but its reliability, its availability, and its business continuity.

You don't want your critical data to be on some cloud computer that abruptly disappears because its owner goes bankrupt. You don't want the company you're using to be sold to your direct competitor. You don't want the company to cut corners, without warning, because times are tight. Or raise its prices and then refuse to let you have your data back. These things can happen with software vendors, but the results aren't as drastic.

There are two different types of cloud computing customers. The first only pays a nominal fee for these services—and uses them for free in exchange for ads: e.g., Gmail and Facebook. These customers have no leverage with their outsourcers. You can lose everything. Companies like Google and Amazon won't spend a lot of time caring. The second type of customer pays considerably for these services: to salesforce.com, MessageLabs, managed network companies, and so on. These customers have more leverage, providing they write their service contracts correctly. Still, nothing is guaranteed.

Trust is a concept as old as humanity, and the solutions are the same as they have always been. Be careful who you trust, be careful what you trust them with, and be careful how much you trust them. Outsourcing is the future of computing. Eventually we'll get this right, but you don't want to be a casualty along the way.

Is Perfect Access Control Possible?

Originally published in Information Security*, September 2009*

This essay appeared as the second half of a point/counterpoint with Marcus Ranum.

Access control is difficult in an organizational setting. On one hand, every employee needs enough access to do his job. On the other hand, every time you give an employee more access, there's more risk: he could abuse that access, or lose information he has access to, or be socially engineered into giving that access to a malfeasant. So a smart, risk-conscious organization will give each employee the exact level of access he needs to do his job, and no more.

Over the years, there's been a lot of work put into role-based access control. But despite the large number of academic papers and high-profile security products, most organizations don't implement it—at all—with the predictable security problems as a result.

Regularly we read stories of employees abusing their database access-control privileges for personal reasons: medical records, tax records, passport records, police records. NSA eavesdroppers spy on their wives and girlfriends. Departing employees take corporate secrets.

A spectacular access control failure occurred in the UK in 2007. An employee of Her Majesty's Revenue & Customs had to send a couple of thousand sample records from a database on all children in the country to National Audit Office. But it was easier for him to copy the entire database of 25 million people onto a couple of disks and put it in the mail than it was to select out just the records needed. Unfortunately, the discs got lost in the mail, and the story was a huge embarrassment for the government.

Eric Johnson at Dartmouth's Tuck School of Business has been studying the problem, and his results won't startle anyone who has thought about it at all. RBAC is very hard to implement correctly. Organizations generally don't even know who has what role. The employee doesn't know, the boss doesn't know—and these days the employee might have more than one boss—and senior management certainly doesn't know. There's a reason RBAC came out of the military; in that world, command structures are simple and well-defined.

Even worse, employees' roles change all the time—Johnson chronicled one business group of 3,000 people that made 1,000 role changes in just three months—and it's often not obvious what information an employee needs until he actually needs it. And information simply isn't that granular. Just as it's much easier to give someone access to an entire file cabinet than to only the particular files he needs, it's much easier to give someone access to an entire database than only the particular records he needs.

This means that organizations either over-entitle or under-entitle employees. But since getting the job done is more important than anything else, organizations tend to over-entitle. Johnson estimates that 50 percent to 90 percent of employees are over-entitled in large organizations. In the uncommon instance where an employee needs access to something he normally doesn't have, there's generally some process for him to get it. And access is almost never revoked once it's been granted. In large formal organizations, Johnson was able to predict how long an employee had worked there based on how much access he had.

Clearly, organizations can do better. Johnson's current work involves building access-control systems with easy self-escalation, audit to make sure that power isn't abused, violation penalties (Intel, for example, issues "speeding tickets" to violators), and compliance rewards. His goal is to implement incentives and controls that manage access without making people too risk-averse.

In the end, a perfect access control system just isn't possible; organizations are simply too chaotic for it to work. And any good system will allow a certain number of access control violations, if they're made in good faith by people just trying to do their jobs. The "speeding ticket" analogy is better than it looks: we post limits of 55 miles per hour, but generally don't start ticketing people unless they're going over 70.

News Media Strategies for Survival for Journalists

Originally published in Twin Cities Daily Planet, *November 14, 2009*

Those of us living through the Internet-caused revolution in journalism can't see what's going to come out the other side: how readers will interact with journalism, what the sources of journalism will be, how journalists will make money. All we do know is that mass-market journalism is hurting, badly, and may not survive. And that we have no idea how to thrive in this new world of digital media.

I have five pieces of advice to those trying to survive and wanting to thrive: based both on experiences as a successful Internet pundit and blogger, and my observations of others, successful and unsuccessful. I'll talk about writing, but everything I say applies to audio and video as well.

One, be interesting. Yes, that's obvious. But the scale is different now. It used to be you could be interesting in aggregate; a few interesting articles or features could carry an entire publication. Now every single piece of writing has to be interesting; otherwise, it won't get read, passed around, or linked to. Have something to say. Pick a niche you can become known for.

Two, be entertaining. Interesting isn't enough; you have to entertain people as well. Internet readers live in a world where millions of things are constantly vying for their attention. Only the best individual pieces of content thrive in

this environment. Often, "best" means "most entertaining." Opinions are dime a dozen on the Internet; you need to make sure yours are worth your readers' time.

Three, be engaging. Readers want to be engaged. They want to be part of a community. They want to engage, with each other as well as with you, on their own terms. Engagement might involve comment or discussion areas, or ways people can follow your work. Anything that limits engagement inhibits community. What this means depends on context; sometimes you have to allow community to develop naturally, even if it's in ways you don't like. Sometimes you need to censor off-topic comments to prevent hateful or annoying commenters from driving others away. In general, though, you should allow anonymous comments. You should make your interface as easy as possible to use. You should reply to your readers. And you shouldn't treat your readers solely as marketing opportunities. The more your writing fosters engagement, the more popular it will be.

Four, be available. Readers need to be able to interact with your writing on their own terms. This means you can't make it difficult for them to find and link to your content. Make sure your content is accessible by any and every Internet device out there. Never take your old writing off the Internet. Never change your URLs. Never make it hard for them to find or link to a URL . Never put your writing behind a paywall. You're part of an ecosystem now; fail to play by the rules and you quickly become isolated.

Five, be agile. The Internet changes all the time; what's true today might not be true in two years. Don't lock yourself in to a particular look, or a particular web technology. Simple interfaces are better than flashy complicated ones; I don't care what your ad agency tells you. Agility applies to making money, too. We have no idea what financial models will thrive in the future, but it seems likely that it will be a portfolio of different things. You'll be more likely to write for different publications. You'll be more likely to figure out cross subsidies, so that some things pay for the others. I have a free blog and a free monthly newsletter, and charge for books, speaking engagements, and consulting. Your mix will be different. If you're lucky, everything you do will augment everything else.

Revolutions are scary times. The old crumbles around us, and we have no idea what—if anything—will be built on its ruins. Remember, though, that human nature doesn't change. People will always gravitate to the interesting, entertaining, engaging, and available, and the agile will be the first on the scene.

Security and Function Creep

Originally published in IEEE Security & Privacy, *January/February 2010*

Security is rarely static. Technology changes the capabilities of both security systems and attackers. But there's something else that changes security's cost/benefit trade-off: how the underlying systems being secured are used. Far too often we build security for one purpose, only to find it being used for another purpose—one it wasn't suited for in the first place. And then the security system has to play catch-up.

Take driver's licenses, for example. Originally designed to demonstrate a credential—the ability to drive a car—they looked like other credentials: medical licenses or elevator certificates of inspection. They were wallet-sized, of course, but they didn't have much security associated with them. Then, slowly, driver's licenses took on a second application: they became age-verification tokens in bars and liquor stores. Of course the security wasn't up to the task—teenagers can be extraordinarily resourceful if they set their minds to it—and over the decades driver's licenses got photographs, tamper-resistant features (once, it was easy to modify the birth year), and technologies that made counterfeiting harder. There was little value in counterfeiting a driver's license, but a lot of value in counterfeiting an age-verification token.

Today, US driver's licenses are taking on yet another function: security against terrorists. The Real ID Act—the government's attempt to make driver's licenses even more secure—has nothing to do with driving or even with buying alcohol, and everything to do with trying to make that piece of plastic an effective way to verify that someone is not on the terrorist watch list. Whether this is a good idea, or actually improves security, is another matter entirely.

You can see this kind of function creep everywhere. Internet security systems designed for informational Web sites are suddenly expected to provide security for banking Web sites. Security systems that are good enough to protect cheap commodities from being stolen are suddenly ineffective once the price of those commodities rises high enough. Application security systems, designed for locally owned networks, are expected to work even when the application is moved to a cloud computing environment. And cloud computing security, designed for the needs of corporations, is expected to be suitable for government applications as well—maybe even military applications.

Sometimes it's obvious that security systems designed for one environment won't work in another. We don't arm our soldiers the same way we arm our policemen, and we can't take commercial vehicles and easily turn them into ones outfitted for the military. We understand that we might need to upgrade our home security system if we suddenly come into possession of a bag of diamonds. Yet many think the same security that protects our home computers will also protect voting machines, and the same operating systems that run our businesses are suitable for military uses.

But these are all conscious decisions, and we security professionals often know better. The real problems arise when the changes happen in the background, without any conscious thought. We build a network security system that's perfectly adequate for the threat and—like a driver's license becoming an age-verification token—the network accrues more and more functions. But because it has already been pronounced "secure," we can't get any budget to re-evaluate and improve the security until after the bad guys have figured out the vulnerabilities and exploited them.

I don't like having to play catch-up in security, but we seem doomed to keep doing so.

Weighing the Risk of Hiring Hackers

Originally published in Information Security, *June 2010*

This essay previously appeared as the first half of a point-counterpoint with Marcus Ranum.

Any essay on hiring hackers quickly gets bogged down in definitions. What is a hacker, and how is he different from a cracker? I have my own definitions, but I'd rather define the issue more specifically: Would you hire someone convicted of a computer crime to fill a position of trust in your computer network? Or, more generally, would you hire someone convicted of a crime for a job related to that crime?

The answer, of course, is "it depends." It depends on the specifics of the crime. It depends on the ethics involved. It depends on the recidivism rate of the type of criminal. It depends a whole lot on the individual.

Would you hire a convicted pedophile to work at a day care center? Would you hire Bernie Madoff to manage your investment fund? The answer is almost certainly no to those two—but you might hire a convicted bank robber to

consult on bank security. You might hire someone who was convicted of false advertising to write ad copy for your next marketing campaign. And you might hire someone who ran a chop shop to fix your car. It depends on the person and the crime.

It can get even murkier. Would you hire a CIA-trained assassin to be a bodyguard? Would you put a general who led a successful attack in charge of defense? What if they were both convicted of crimes in whatever country they were operating in? There are different legal and ethical issues, to be sure, but in both cases the people learned a certain set of skills regarding offense that could be transferable to defense.

Which brings us back to computers. Hacking is primarily a mindset: a way of thinking about security. Its primary focus is in attacking systems, but it's invaluable to the defense of those systems as well. Because computer systems are so complex, defending them often requires people who can think like attackers.

Admittedly, there's a difference between thinking like an attacker and acting like a criminal, and between researching vulnerabilities in fielded systems and exploiting those vulnerabilities for personal gain. But there is a huge variability in computer crime convictions, and—at least in the early days—many hacking convictions were unjust and unfair. And there's also a difference between someone's behavior as a teenager and his behavior later in life. Additionally, there might very well be a difference between someone's behavior before and after a hacking conviction. It all depends on the person.

An employer's goal should be to hire moral and ethical people with the skill set required to do the job. And while a hacking conviction is certainly a mark against a person, it isn't always grounds for complete non-consideration.

"We don't hire hackers" and "we don't hire felons" are coarse generalizations, in the same way that "we only hire people with this or that security certification" is. They work—you're less likely to hire the wrong person if you follow them—but they're both coarse and flawed. Just as all potential employees with certifications aren't automatically good hires, all potential employees with hacking convictions aren't automatically bad hires. Sure, it's easier to hire people based on things you can learn from checkboxes, but you won't get the best employees that way. It's far better to look at the individual, and put those check boxes into context. But we don't always have time to do that.

Last winter, a Minneapolis attorney who works to get felons a fair shake after they served their time told of a sign he saw: "Snow shovelers wanted. Felons need not apply." It's not good for society if felons who have served their time can't even get jobs shoveling snow.

Should Enterprises Give In to IT Consumerization at the Expense of Security?

Originally published in Information Security, *September 2010*

This essay appeared as the second half of a point/counterpoint with Marcus Ranum.

If you're a typical wired American, you've got a bunch of tech tools you like and a bunch more you covet. You have a cell phone that can easily text. You've got a laptop configured just the way you want it. Maybe you have a Kindle for reading, or an iPad. And when the next new thing comes along, some of you will line up on the first day it's available.

So why can't work keep up? Why are you forced to use an unfamiliar, and sometimes outdated, operating system? Why do you need a second laptop, maybe an older and clunkier one? Why do you need a second cell phone with a new interface, or a BlackBerry, when your phone already does e-mail? Or a second BlackBerry tied to corporate e-mail? Why can't you use the cool stuff you already have?

More and more companies are letting you. They're giving you an allowance and allowing you to buy whatever laptop you want, and to connect into the corporate network with whatever device you choose. They're allowing you to use whatever cell phone you have, whatever portable e-mail device you have, whatever you personally need to get your job done. And the security office is freaking.

You can't blame them, really. Security is hard enough when you have control of the hardware, operating system and software. Lose control of any of those things, and the difficulty goes through the roof. How do you ensure that the employee devices are secure, and have up-to-date security patches? How do you control what goes on them? How do you deal with the tech support issues when they fail? How do you even begin to manage this logistical nightmare? Better to dig your heels in and say "no."

But security is on the losing end of this argument, and the sooner it realizes that, the better.

The meta-trend here is consumerization: cool technologies show up for the consumer market before they're available to the business market. Every corporation is under pressure from its employees to allow them to use these

new technologies at work, and that pressure is only getting stronger. Younger employees simply aren't going to stand for using last year's stuff, and they're not going to carry around a second laptop. They're either going to figure out ways around the corporate security rules, or they're going to take another job with a more trendy company. Either way, senior management is going to tell security to get out of the way. It might even be the CEO, who wants to get to the company's databases from his brand new iPad, driving the change. Either way, it's going to be harder and harder to say no.

At the same time, cloud computing makes this easier. More and more, employee computing devices are nothing more than dumb terminals with a browser interface. When corporate e-mail is all webmail, corporate documents are all on GoogleDocs, and when all the specialized applications have a web interface, it's easier to allow employees to use any up-to-date browser. It's what companies are already doing with their partners, suppliers, and customers.

Also on the plus side, technology companies have woken up to this trend and—from Microsoft and Cisco on down to the startups—are trying to offer security solutions. Like everything else, it's a mixed bag: some of them will work and some of them won't, most of them will need careful configuration to work well, and few of them will get it right. The result is that we'll muddle through, as usual.

Security is always a tradeoff, and security decisions are often made for non-security reasons. In this case, the right decision is to sacrifice security for convenience and flexibility. Corporations want their employees to be able to work from anywhere, and they're going to have loosened control over the tools they allow in order to get it.

The Vulnerabilities Market and the Future of Security

Originally published in Forbes*, May 30, 2012*

Recently, there have been several articles about the new market in zero-day exploits: new and unpatched computer vulnerabilities. It's not just software companies, who sometimes pay bounties to researchers who alert them of security vulnerabilities so they can fix them. And it's not only criminal organizations that pay for vulnerabilities they can exploit. Now there are governments,

and companies who sell to governments, who buy vulnerabilities with the intent of keeping them secret so they can exploit them.

This market is larger than most people realize, and it's becoming even larger. Forbes recently published a price list for zero-day exploits, along with the story of a hacker who received $250K from "a US government contractor." (At first I didn't believe the story or the price list, but I have been convinced that they both are true.) Forbes published a profile of a company called Vupen, whose business is selling zero-day exploits. Other companies doing this range from startups like Netragard and Endgame to large defense contractors like Northrop Grumman, General Dynamics, and Raytheon.

This is very different than in 2007, when researcher Charlie Miller wrote about his attempts to sell zero-day exploits; and a 2010 survey implied that there wasn't much money in selling zero days. The market has matured substantially in the past few years.

This new market perturbs the economics of finding security vulnerabilities. And it does so to the detriment of us all.

I've long argued that the process of finding vulnerabilities in software systems increases overall security. This is because the economics of vulnerability hunting favored disclosure. As long as the principal gain from finding a vulnerability was notoriety, publicly disclosing vulnerabilities was the only obvious path. In fact, it took years for our industry to move from a norm of full-disclosure—announcing the vulnerability publicly and damn the consequences—to something called "responsible disclosure": giving the software vendor a head start in fixing the vulnerability. Changing economics is what made the change stick: instead of just hacker notoriety, a successful vulnerability finder could land some lucrative consulting gigs, and being a responsible security researcher helped. But regardless of the motivations, a disclosed vulnerability is one that—at least in most cases—is patched. And a patched vulnerability makes us all more secure.

This is why the new market for vulnerabilities is so dangerous; it results in vulnerabilities remaining secret and unpatched. That it's even more lucrative than the public vulnerabilities market means that more hackers will choose this path. And unlike the previous reward of notoriety and consulting gigs, it gives software programmers within a company the incentive to deliberately create vulnerabilities in the products they're working on—and then secretly sell them to some government agency.

No commercial vendors perform the level of code review that would be necessary to detect, and prove mal-intent for, this kind of sabotage.

Even more importantly, the new market for security vulnerabilities results in a variety of government agencies around the world that have a strong interest in those vulnerabilities remaining unpatched. These range from law-enforcement agencies like the FBI and the German police who are trying to build targeted Internet surveillance tools, to intelligence agencies like the NSA who are trying to build mass Internet surveillance tools, to military organizations who are trying to build cyber-weapons.

All of these agencies have long had to wrestle with the choice of whether to use newly discovered vulnerabilities to protect or to attack. Inside the NSA, this was traditionally known as the "equities issue," and the debate was between the COMSEC (communications security) side of the NSA and the SIGINT (signals intelligence) side. If they found a flaw in a popular cryptographic algorithm, they could either use that knowledge to fix the algorithm and make everyone's communications more secure, or they could exploit the flaw to eavesdrop on others—while at the same time allowing even the people they wanted to protect to remain vulnerable. This debate raged through the decades inside the NSA. From what I've heard, by 2000, the COMSEC side had largely won, but things flipped completely around after 9/11.

The whole point of disclosing security vulnerabilities is to put pressure on vendors to release more secure software. It's not just that they patch the vulnerabilities that are made public—the fear of bad press makes them implement more secure software development processes. It's another economic process; the cost of designing software securely in the first place is less than the cost of the bad press after a vulnerability is announced plus the cost of writing and deploying the patch. I'd be the first to admit that this isn't perfect—there's a lot of very poorly written software still out there—but it's the best incentive we have.

We've always expected the NSA, and those like them, to keep the vulnerabilities they discover secret. We have been counting on the public community to find and publicize vulnerabilities, forcing vendors to fix them. With the rise of these new pressures to keep zero-day exploits secret, and to sell them for exploitation, there will be even less incentive on software vendors to ensure the security of their products.

As the incentive for hackers to keep their vulnerabilities secret grows, the incentive for vendors to build secure software shrinks. As a recent EFF essay put it, this is "security for the 1%." And it makes the rest of us less safe.

So You Want to Be a Security Expert

Originally published in Krebs on Security, *July 12, 2012*

This essay originally appeared as part of a series of advice columns on how to break into the field of security.

I regularly receive e-mail from people who want advice on how to learn more about computer security, either as a course of study in college or as an IT person considering it as a career choice.

First, know that there are many subspecialties in computer security. You can be an expert in keeping systems from being hacked, or in creating unhackable software. You can be an expert in finding security problems in software, or in networks. You can be an expert in viruses, or policies, or cryptography. There are many, many opportunities for many different skill sets. You don't have to be a coder to be a security expert.

In general, though, I have three pieces of advice to anyone who wants to learn computer security.

Study. Studying can take many forms. It can be classwork, either at universities or at training conferences like SANS and Offensive Security. (These are good self-starter resources.) It can be reading; there are a lot of excellent books out there—and blogs—that teach different aspects of computer security out there. Don't limit yourself to computer science, either. You can learn a lot by studying other areas of security, and soft sciences like economics, psychology, and sociology.

Do. Computer security is fundamentally a practitioner's art, and that requires practice. This means using what you've learned to configure security systems, design new security systems, and—yes—break existing security systems. This is why many courses have strong hands-on components; you won't learn much without it.

Show. It doesn't matter what you know or what you can do if you can't demonstrate it to someone who might want to hire you. This doesn't just mean sounding good in an interview. It means sounding good on mailing lists and in blog comments. You can show your expertise by making podcasts and writing your own blog. You can teach seminars at your local user group meetings. You can write papers for conferences, or books.

I am a fan of security certifications, which can often demonstrate all of these things to a potential employer quickly and easily.

I've really said nothing here that isn't also true for a gazillion other areas of study, but security also requires a particular mindset—one I consider essential for success in this field. I'm not sure it can be taught, but it certainly can be encouraged. "This kind of thinking is not natural for most people. It's not natural for engineers. Good engineering involves thinking about how things can be made to work; the security mindset involves thinking about how things can be made to fail. It involves thinking like an attacker, an adversary or a criminal. You don't have to exploit the vulnerabilities you find, but if you don't see the world that way, you'll never notice most security problems." This is especially true if you want to design security systems and not just implement them. Remember Schneier's Law: "Any person can invent a security system so clever that she or he can't think of how to break it." The only way your designs are going to be trusted is if you've made a name for yourself breaking other people's designs.

One final word about cryptography. Modern cryptography is particularly hard to learn. In addition to everything above, it requires graduate-level knowledge in mathematics. And, as in computer security in general, your prowess is demonstrated by what you can break. The field has progressed a lot since I wrote this guide and self-study cryptanalysis course a dozen years ago, but they're not bad places to start.

When It Comes to Security, We're Back to Feudalism

Originally published in Wired*, November 26, 2012*

Some of us have pledged our allegiance to Google: We have Gmail accounts, we use Google Calendar and Google Docs, and we have Android phones. Others have pledged allegiance to Apple: We have Macintosh laptops, iPhones, and iPads; and we let iCloud automatically synchronize and back up everything. Still others of us let Microsoft do it all. Or we buy our music and e-books from Amazon, which keeps records of what we own and allows downloading to a Kindle, computer, or phone. Some of us have pretty much abandoned e-mail altogether. . . for Facebook.

These vendors are becoming our feudal lords, and we are becoming their vassals. We might refuse to pledge allegiance to all of them—or to a particular one we don't like. Or we can spread our allegiance around. But either way, it's becoming increasingly difficult to not pledge allegiance to at least one of them.

Feudalism provides security. Classical medieval feudalism depended on overlapping, complex, hierarchical relationships. There were oaths and obligations: a series of rights and privileges. A critical aspect of this system was protection: vassals would pledge their allegiance to a lord, and in return, that lord would protect them from harm.

Of course, I'm romanticizing here; European history was never this simple, and the description is based on stories of that time, but that's the general model.

And it's this model that's starting to permeate computer security today.

I Pledge Allegiance to the United States of Convenience

Traditional computer security centered around users. Users had to purchase and install anti-virus software and firewalls, ensure their operating system and network were configured properly, update their software, and generally manage their own security.

This model is breaking, largely due to two developments:

1. New Internet-enabled devices where the vendor maintains more control over the hardware and software than we do—like the iPhone and Kindle; and
2. Services where the host maintains our data for us—like Flickr and Hotmail.

Now, we users must trust the security of these hardware manufacturers, software vendors, and cloud providers.

We choose to do it because of the convenience, redundancy, automation, and shareability. We like it when we can access our e-mail anywhere, from any computer. We like it when we can restore our contact lists after we've lost our phones. We want our calendar entries to automatically appear on all of our devices. These cloud storage sites do a better job of backing up our photos and files than we would manage by ourselves; Apple does a great job keeping malware out of its iPhone apps store.

In this new world of computing, we give up a certain amount of control, and in exchange we trust that our lords will both treat us well and protect us from harm. Not only will our software be continually updated with the newest and coolest functionality, but we trust it will happen without our being overtaxed by fees and required upgrades. We trust that our data and devices won't be exposed to hackers, criminals, and malware. We trust that governments won't be allowed to illegally spy on us.

Trust is our only option. In this system, we have no control over the security provided by our feudal lords. We don't know what sort of security methods they're using, or how they're configured. We mostly can't install our own security products on iPhones or Android phones; we certainly can't install them on Facebook, Gmail, or Twitter. Sometimes we have control over whether or not to accept the automatically flagged updates—iPhone, for example—but we rarely know what they're about or whether they'll break anything else. (On the Kindle, we don't even have that freedom.)

The Good, the Bad, and the Ugly

I'm not saying that feudal security is all bad. For the average user, giving up control is largely a good thing. These software vendors and cloud providers do a lot better job of security than the average computer user would. Automatic cloud backup saves a lot of data; automatic updates prevent a lot of malware. The network security at any of these providers is better than that of most home users.

Feudalism is good for the individual, for small startups, and for medium-sized businesses that can't afford to hire their own in-house or specialized expertise. Being a vassal has its advantages, after all.

For large organizations, however, it's more of a mixed bag. These organizations are used to trusting other companies with critical corporate functions: They've been outsourcing their payroll, tax preparation, and legal services for decades. But IT regulations often require audits. Our lords don't allow vassals to audit them, even if those vassals are themselves large and powerful.

Yet feudal security isn't without its risks.

Our lords can make mistakes with security, as recently happened with Apple, Facebook, and Photobucket. They can act arbitrarily and capriciously, as Amazon did when it cut off a Kindle user for living in the wrong country. They tether us like serfs; just try to take data from one digital lord to another.

Ultimately, they will always act in their own self-interest, as companies do when they mine our data in order to sell more advertising and make more

money. These companies own us, so they can sell us off—again, like serfs—to rival lords. . . or turn us into the authorities.

Historically, early feudal arrangements were ad hoc, and the more powerful party would often simply renege on his part of the bargain. Eventually, the arrangements were formalized and standardized: both parties had rights and privileges (things they could do) as well as protections (things they couldn't do to each other).

Today's Internet feudalism, however, is ad hoc and one-sided. We give companies our data and trust them with our security, but we receive very few assurances of protection in return, and those companies have very few restrictions on what they can do.

This needs to change. There should be limitations on what cloud vendors can do with our data; rights, like the requirement that they delete our data when we want them to; and liabilities when vendors mishandle our data.

Like everything else in security, it's a trade-off. We need to balance that trade-off. In Europe, it was the rise of the centralized state and the rule of law that undermined the ad hoc feudal system; it provided more security and stability for both lords and vassals. But these days, government has largely abdicated its role in cyberspace, and the result is a return to the feudal relationships of yore.

Perhaps instead of hoping that our Internet-era lords will be sufficiently clever and benevolent—or putting our faith in the Robin Hoods who block phone surveillance and circumvent DRM systems—it's time we step in in our role as governments (both national and international) to create the regulatory environments that protect us vassals (and the lords as well). Otherwise, we really are just serfs.

You Have No Control Over Security on the Feudal Internet

Originally published in Harvard Business Review, *June 6, 2013*

Facebook regularly abuses the privacy of its users. Google has stopped supporting its popular RSS feeder. Apple prohibits all iPhone apps that are political or sexual. Microsoft might be cooperating with some governments to spy on Skype calls, but we don't know which ones. Both Twitter and LinkedIn

have recently suffered security breaches that affected the data of hundreds of thousands of their users.

If you've started to think of yourself as a hapless peasant in a *Game of Thrones* power struggle, you're more right than you may realize. These are not traditional companies, and we are not traditional customers. These are feudal lords, and we are their vassals, peasants, and serfs.

Power has shifted in IT, in favor of both cloud-service providers and closed-platform vendors. This power shift affects many things, and it profoundly affects security.

Traditionally, computer security was the user's responsibility. Users purchased their own antivirus software and firewalls, and any breaches were blamed on their inattentiveness. It's kind of a crazy business model. Normally we expect the products and services we buy to be safe and secure, but in IT we tolerated lousy products and supported an enormous aftermarket for security.

Now that the IT industry has matured, we expect more security "out of the box." This has become possible largely because of two technology trends: cloud computing and vendor-controlled platforms. The first means that most of our data resides on other networks: Google Docs, Salesforce.com, Facebook, Gmail. The second means that our new Internet devices are both closed and controlled by the vendors, giving us limited configuration control: iPhones, ChromeBooks, Kindles, Blackberries. Meanwhile, our relationship with IT has changed. We used to use our computers to do things. We now use our vendor-controlled computing devices to go places. All of these places are owned by someone.

The new security model is that someone else takes care of it—without telling us any of the details. I have no control over the security of my Gmail or my photos on Flickr. I can't demand greater security for my presentations on Prezi or my task list on Trello, no matter how confidential they are. I can't audit any of these cloud services. I can't delete cookies on my iPad or ensure that files are securely erased. Updates on my Kindle happen automatically, without my knowledge or consent. I have so little visibility into the security of Facebook that I have no idea what operating system they're using.

There are a lot of good reasons why we're all flocking to these cloud services and vendor-controlled platforms. The benefits are enormous, from cost to convenience to reliability to security itself. But it is inherently a feudal relationship. We cede control of our data and computing platforms to these companies and trust that they will treat us well and protect us from harm. And if we pledge complete allegiance to them—if we let them control our email

and calendar and address book and photos and everything—we get even more benefits. We become their vassals; or, on a bad day, their serfs.

There are a lot of feudal lords out there. Google and Apple are the obvious ones, but Microsoft is trying to control both user data and the end-user platform as well. Facebook is another lord, controlling much of the socializing we do on the Internet. Other feudal lords are smaller and more specialized—Amazon, Yahoo, Verizon, and so on—but the model is the same.

To be sure, feudal security has its advantages. These companies are much better at security than the average user. Automatic backup has saved a lot of data after hardware failures, user mistakes, and malware infections. Automatic updates have increased security dramatically. This is also true for small organizations; they are more secure than they would be if they tried to do it themselves. For large corporations with dedicated IT security departments, the benefits are less clear. Sure, even large companies outsource critical functions like tax preparation and cleaning services, but large companies have specific requirements for security, data retention, audit, and so on—and that's just not possible with most of these feudal lords.

Feudal security also has its risks. Vendors can, and do, make security mistakes affecting hundreds of thousands of people. Vendors can lock people into relationships, making it hard for them to take their data and leave. Vendors can act arbitrarily, against our interests; Facebook regularly does this when it changes peoples' defaults, implements new features, or modifies its privacy policy. Many vendors give our data to the government without notice, consent, or a warrant; almost all sell it for profit. This isn't surprising, really; companies should be expected to act in their own self-interest and not in their users' best interest.

The feudal relationship is inherently based on power. In Medieval Europe, people would pledge their allegiance to a feudal lord in exchange for that lord's protection. This arrangement changed as the lords realized that they had all the power and could do whatever they wanted. Vassals were used and abused; peasants were tied to their land and became serfs.

It's the Internet lords' popularity and ubiquity that enable them to profit; laws and government relationships make it easier for them to hold onto power. These lords are vying with each other for profits and power. By spending time on their sites and giving them our personal information—whether through search queries, e-mails, status updates, likes, or simply our behavioral characteristics—we are providing the raw material for that struggle. In this way we are like serfs, toiling the land for our feudal lords. If you don't believe me, try

to take your data with you when you leave Facebook. And when war breaks out among the giants, we become collateral damage.

So how do we survive? Increasingly, we have little alternative but to trust *someone*, so we need to decide who we trust—and who we don't—and then act accordingly. This isn't easy; our feudal lords go out of their way not to be transparent about their actions, their security, or much of anything. Use whatever power you have—as individuals, none; as large corporations, more—to negotiate with your lords. And, finally, don't be extreme in any way: politically, socially, culturally. Yes, you can be shut down without recourse, but it's usually those on the edges that are affected. Not much solace, I agree, but it's something.

On the policy side, we have an action plan. In the short term, we need to keep circumvention—the ability to modify our hardware, software, and data files—legal and preserve net neutrality. Both of these things limit how much the lords can take advantage of us, and they increase the possibility that the market will force them to be more benevolent. The last thing we want is the government—that's us—spending resources to enforce one particular business model over another and stifling competition.

In the longer term, we all need to work to reduce the power imbalance. Medieval feudalism evolved into a more balanced relationship in which lords had responsibilities as well as rights. Today's Internet feudalism is both ad-hoc and one-sided. We have no choice but to trust the lords, but we receive very few assurances in return. The lords have a lot of rights, but few responsibilities or limits. We need to balance this relationship, and government intervention is the only way we're going to get it. In medieval Europe, the rise of the centralized state and the rule of law provided the stability that feudalism lacked. The Magna Carta first forced responsibilities on governments and put humans on the long road toward government by the people and for the people.

We need a similar process to rein in our Internet lords, and it's not something that market forces are likely to provide. The very definition of power is changing, and the issues are far bigger than the Internet and our relationships with our IT providers.