Inflection points come at you without warning and quickly recede out of reach. We *may* be nearing one now. If so, we are now about to play for keeps, and “we” doesn’t mean just us security geeks. If anything, it’s because we security geeks have not worked the necessary miracles already that an inflection point seems to be approaching at high velocity.

Many of us believe and many more of us say that complexity and security are antipodal. This *complexity vs. security* dichotomy is real but not exact; yet it is to some degree measurable, and news from that front is not good. The software industry sells a product that does not naturally wear out and that retains complete fidelity when copied—two characteristics, among others, that separate the digital world from the physical world. To continue to make money from existing customers, a software supplier must sell upgrades, maintenance, or both. Maintenance sells best when a product is unstable or hard to use—the very need for maintenance is an admission of complexity. New features, if they are to compel otherwise happy users to effectively repurchase a product they already have, tend to be at least linear (10 new features) if not geometric (10% new features). Absent perfection, each new feature comes with new failure modes, and features can sometimes interact; therefore, the potential number of failure modes quite naturally can grow faster than the feature count.

DANIEL E. GEER, VERDASYS

Playing for Keeps

Will security threats bring an end to general-purpose computing?

Playing for Keeps

WE'VE BEEN WARNED

As we already knew when Fred Brooks wrote *The Mythical Man-Month* in 1975, an increase in features with each new release of a software product means that as the product grows, the rate of bug finding falls at first but then begins to rise, as shown in figure 1.

Brooks estimates that 20 to 50 percent of all fixes for known bugs introduce unknown bugs. Therefore, rationally speaking, there comes a time at which many bugs should be left permanently in place and fully documented instead of fixed. Excepting the unlikely, obscure, and special case of security flaws that are intentionally introduced into products, security flaws are merely a subset of all unintentional flaws and will thus also rise with system complexity. The difficulty is that for a security bug to be left permanently in place and fully documented paints a bulls-eye on the product and its users. If Brooks's guess is correct—that 20 to 50 percent of all fixes for known bugs introduce unknown bugs—and if leaving a security bug in place is anathema, one can only then conclude

that: 1. The distributed existence of unknown security bugs is a permanent characteristic of the digital world; and 2. the more complex the product, the more likely this is to be so for that product. Add in ubiquity if not monoculture, and you have a problem of some difficulty.

It is hard to make a security product that isn't itself subject to misuse. Indeed, attacks against security products seem to be rising faster than attacks against all software products in the aggregate, thus making security products over-represented, statistically speaking, as attack targets. This is not surprising: Every Mafia chieftain will first try to recruit his competitor's bodyguard before he tries to blow by him; the diseases we fear most are those that defeat our immune system by commandeering it rather than by evading it; to make you into a suicide bomber, I just fill your trunk with explosives when you are not looking.

Security products are, for example, seemingly tailor-made for DoS (denial of service) attacks. Your policy is to disable accounts after three failed logins in a short interval? Very well, I will just fail login for every one of your users (and the bigger you are, the easier it is to find real user names). To defeat guessing attacks, you employ computationally expensive authorization tests? Very well, whatever the compute cost is for you to make a NOGO decision is the cost I can impose on you at will; if my cost

to initiate authorization is less than your cost to deny it, we have that asymmetry of which DoS is made.

To make a product that everyone wants to run (and everyone can run), that product must be dumbed down. This is a nonspecific insult, to be sure, but dumbing down products is and always has been the *sine qua non* of mass markets. That is especially true of software products, since they grow in complexity far faster than their users grow in smarts. Moore's law lets us make computational intensity rise by twice every 18 months, but nobody's users start thinking twice as fast every 18 months.

Source: *The Mythical Man-Month*, by Fred Brooks, Addison-Wesley Professional, 1995

FIG 1

IRRESISTIBLE FORCES

Consumer devices, in the Internet age, derive a significant percentage of their perceived value from how easily connected they are. Connecting everything with everything can only continue; the laboratory makes it so. Moore's law doubles CPU per dollar every 18 months, but storage doubles at 12, and bandwidth faster still at perhaps nine. This says that the economically optimal electronic device, including the computer, changes over time in the direction of more data moving much more often. If that 18:12:9 ratio is both reasonably accurate and holds for a decade, you get two orders of magnitude in CPU but three in storage and four in bandwidth. If we hold computer design constant, a decade of such growth puts 10 times as much data in front of the CPU as there is today, but that data is simultaneously 10 times as mobile in the aggregate. The Verizon FiOS fiber-optic broadband service, for example, serves as a kind of verification on the ground of a future that is more data rich but even more data mobile.

If this prediction of a data-rich, data-mobile future is true, then you do not have to be a professional paranoid to imagine that data eclipses CPUs as the primary focus of attacks carried out over ever-faster networks. Perhaps that is already so, in that a lost laptop is a minor financial cost to replace but the data it contained is neither minor nor merely financial. What makes this an inflection point is itself an example of the difference between the digital world and the physical one: Stolen data leaves the source entirely operable; you will not immediately notice that it is missing (as you would a car stolen from your driveway), but you will discover data was stolen only if it is used in a way that is visible to you. There is no exclusion principle here (if I have your car, you don't; but if I have your data, so do you).

SO THAT'S WHY WE'RE HERE?

Let's review: Complexity rises because the self-interest of software vendors demands it. The orders-of-magnitude advances coming out of the laboratory make computers and computer-like devices ever more available to ever more people. The ratio of (people) skill to (computer) horsepower is falling quickly, and will not fall less quickly. As prices for hardware fall, it is data that assumes the preponderance of value. Thieves robbed banks when that was where the money was; now they rob data, and it would be surprising if they didn't. As every marketeer and every intelligence officer knows, data fusion increases the value of what data you do have. Our opposition is intelligent in this regard, too.

One might suggest that the public doesn't know how fragile this all is, but that state of innocence is passing one way or another. The more helpless the public feels, the more it wants to be protected. This is, in fact, a central theme of this essay—the public feels helpless, and feeling helpless damps out initiative. This is true not only of what we speak of here.

PARTY TIME IS OVER

When attackers assume little if any risk to make an attack, they will attack with abandon. When attackers can use automation, they will attack with vigor. When attackers' fundamental operational costs are a mere fraction of defenders' fundamental operational costs, the attackers can win the arms race. When attackers can mount assaults without warning signs, defenders must always be on high alert. All of these things can be obtained in the digital arena, and when that happens, the only strategy is worst-case preemption. This is true in the world of terrorism but truer yet in the digital world.

The only alternative to the problem of **complexity vs. security** is to make computing not be so general purpose.

Preemption requires intelligence, intelligence requires surveillance, and surveillance requires mechanisms for doing so that do not depend on the volition or the sentience of those under surveillance. The public is demanding protection that it feels unable to accomplish itself. The public, at least in the United States, is inured to the idea that all bad outcomes are the fault of someone to whom liability can be assigned. The public is wrong in principle (take care of yourself) but right in practice (assign risk to those most capable of thwarting it). These forces conspire to put the duty of surveillance on the upstream end, to assign the duty (and the liability) of protection to those with the most resources regardless of whether it is fair.

We've done this before—Regulation Z of the Truth in Lending Act of 1968 says that the most a consumer can lose from misuse of a credit card is \$50. The consumer can be an idiot, but can't lose more than \$50. Consum-

Playing for Keeps

ers are, in fact, not encouraged to self-protect by such a limit—quite the opposite (and \$50 in 1968 would be \$275 today). No, if there is to be a preemption, the intelligence it requires will be based on a duty of surveillance that is assigned to various “deep pockets.” The countermeasures, in other words, are not risk-sensitive to where the risk naturally lies but risk-sensitive to where it is assigned. Look out side effects, here we come.

If the future holds more data and that data is both more mobile and the only real store of value, then we’ve come at last to Grace Murray Hopper’s 1987 prediction:

“Some day, on the corporate balance sheet, there will be an entry which reads, ‘Information’; for in most cases the information is more valuable than the hardware which processes it.”

And if you think that the data doesn’t have balance-sheet relevance, calculate the liquidation value of an information-only company—say, Fair Isaac. If we are to surveil something to protect the asset (data value), then we have to select a unit of observation. That may be the only open question: Is the unit of observation one data item or is the unit of observation one person? Do we build the infrastructure to surveil data or people? We have aspects of that in place already; what is usually called DRM (digital rights management) sort of surveils data, but is hampered by data residing in a hostile location.

Closer to the point, everyone who carries a cellphone in the U.S. is under surveillance by the imposed requirement of instant location for emergency (911) services. So, what do we want as a unit of surveillance? Remembering the power of fusing data from one surveillance system with another, think carefully before you answer. Think about whether you want surveillance at all. Think about what you are willing to trade for safety or, frankly, that obliviousness that seems to be synonymous with a feeling of safety. My own family is quite naturally tired of me and this issue, but someone who should know better says, “Privacy doesn’t matter to me. I live a good life. I have nothing to hide.”

This is not meant to be a diatribe about privacy, however much the word *surveillance* may imply that it is. It is a question of what the future of computing is. Here is the question that I am really asking: If, by some miracle, my

friends and neighbors decide that the safety they want is not available without a level of surveillance they can’t knowingly accept, what then?

I suppose they can accept surveillance they don’t know about (“Just keep me safe, but don’t tell me about it”). I suppose they can decide to swear off the Internet in some sense—after all, in the real world no one wants to live in a part of town where every sociopath is your next-door neighbor. I suppose they can refuse to buy cellphones with GPS and insert the battery only when they want to make a call. But I doubt it.

YOU ARE TOO DIFFERENT TO MATTER

We here in the geek world, the people who actually find it pleasurable to read *Queue*, are and always will be a minority. There are not and never will be enough of us to make the kinds of things we know a part of “literacy.” We cannot use ourselves as models for society, though to the extent that everyone within the sound of my voice is almost surely the systems administrator for their extended family, we may grasp what other people want. If nothing else, it is already true that there are nearly no software products that both matter and that any one person fully understands right down to the iron, at least no software products that matter to the public. Sure, my colleague Hobbit knows *ncat* because he wrote it, but who can explain its usefulness to that man in the street? Andreesen probably understood Mosaic at some point, Gutmann knows *cryptlib*, and Dingledine knows *tor*, but how much of Windows does Cutler still grok? Ditto Linux and Torvalds? And why am I saying this?

The snowballing complexity our software industry calls progress is generating ever-more subtle flaws. It cannot do otherwise. It will not do otherwise. This is physics, not human failure. Sure, per unit volume of code, it may be getting better. But the amount of code you can run per unit of time or for X dollars is growing at geometric rates. Therefore, for constant risk the goodness of our software has to be growing at geometric rates just to stay even. The Red Queen was right: You have to run faster and faster to stay in the same place. That’s true competitively, and it is true in security terms: Symantec’s Internet Threat Reports show rising vulnerability detection rates (see figure 2).

While staring at that graph in figure 2, remember that in this same time period there have been spectacular increases in spending at every software vendor to avoid vulnerabilities—and still the curve goes up. We are evidently not running hard enough because we are not staying in the same place. Enterprises know this. It is even a high goal for some; the absolutely most daunting RFP

any bank ever sent me was no more than this:

1. Come do your worst to us.
2. Tell us, in numbers, how secure we are.
3. From then on, send us engineering change orders that keep us at constant risk.

Frankly, I slunk away with my tail between my legs.

WHERE THIS GOES

So what's the point? The only alternative to the problem of *complexity vs. security* is to make computing not be so general purpose, to get the complexity out by creating appliances instead. Corporate America is trying hard to do this: Every lock-down script, every standard build, every function turned off by default is an attempt to reduce the attack surface by reducing the generality. The generality is where the complexity lives, in exactly the same way that the Perl mantra—*there's always another way*—is why correctness for Perl can be no more than “Did it work?” (Perl hackers, this is not about you.) This is what is driving the “virtualization” meme.

Virtualization damps complexity risk by replacing the general-purpose computer with purpose-built appliances. If momentum is mass times velocity ($p = mv$), then virtualization has a lot of momentum already, and it will get more. Microsoft has a head of steam up in this direction; this is why it bought Connectix. Several of the proverbial

big banks in New York are converting their trading floors so that the desktops only have displays, and every app you see there is effectively an appliance because it is running solo in some virtual machine on big iron in distant, redundant data centers. Gartner has now prophesized that Vista is the last general-purpose operating system that Redmond will ever produce (and if Gartner says it, it's officially safe for every CIO to say so, too).

An “appliance strategy” dodges the insecurity that increasingly complex general-purpose computers cannot escape. Whether it is a side effect or a purpose, little virtual machines that are fast to restart also get you high availability by making recovery time near zero. As you probably already know:

$$\text{Availability} = \text{MTBF} / (\text{MTBF} + \text{MTTR})$$

Thus, availability is 100 percent whenever MTBF (mean time between failures) is infinite or when MTTR (mean time to recovery) is zero. Those banks, having spent ten figures (USD) on avoiding failures, have decided to spend no more on MTBF but instead concentrate on MTTR. They'll dedicate one VM to one task and they'll get the fast recovery they need. Others may well copy them, but for those that do, the change will be a hysteresis—once virtualization takes hold, there will be no going back to

the monolithic, general-purpose operating system. To go back, they would have to reabsorb—as a single bolus—all that complexity they had left behind. The level of complexity (and insecurity) besetting today's Internet-connected operating systems is like the frog—it had to be boiled incrementally.

But surveillance remains a clear option, and, if anything, it has much, much more momentum behind it—especially if you count what is going on in the world of physical security, a world with which we digital security folks are supposedly “converging.” Just as the 1990s saw the

New Vulnerabilities per Six-Month Period

FIG 2

Playing for Keeps

commercial world almost entirely catch up to the military world in uses of cryptography, in this decade we catch up to them in traffic analysis. Intrusion systems, firewalls, policy managers, e-mail and URL filters, device-driver code injections, and more (disclaimer: my own company is in this game) are all examples of traffic analysis, broadly defined. Unlike virtualization, surveillance is a layered product on top of what already is present. Surveillance says that the surveillant will protect that which cannot protect itself, and the public absolutely wants to be protected.

YOUR TIME IS ALMOST UP

The people reading *Queue* are not the ones who get to make these decisions. In the first place, the decisions are not made in any one place. And, in any case, when we handed an insecure medium to everyone, we did not do either of the two things that truly mattered: 1. Provide safety, or 2. Obtain informed consent. (I say that as someone who believes you take care of yourself, a restatement of item 2.) I may be being acutely over-optimistic; this may not be a choice between (a) the end of the general-purpose computer versus (b) a surveillance world. It may not be (a) or (b) but rather (c): All of the above.

Let's at least understand where we are. We digerati have given the world fast, free, open transmission to anyone from anyone, and we've handed them a general-purpose device with so many layers of complexity that there is no one who understands it all. Because "you're on your own" won't fly politically, something has to change. Since you don't have to block transmission in order to surveil it, and since general-purpose capabilities in computers are lost on the vast majority of those who use them, the beneficiaries of protection will likely consider surveillance and appliances to be an improvement over risk and complexity. From where they sit, this is true and normal.

While the readers of *Queue* may well appreciate that driving is much more real with a centrifugal advance and a stick shift, try and sell that to the mass market. The general-purpose computer must die or we must put everything under surveillance. Either option is ugly, but "all of the above" would be lights-out for people like me, people like you, people like us. We're playing for keeps now. Q

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

DANIEL E. GEER, Jr., Sc.D. is vice president and chief scientist of Verdasys, Inc. Highlights of his career include the X Window System and Kerberos (1988), the first information security consulting firm on Wall Street (1992), convenor of the first academic conference on electronic commerce (1995), the "Risk Management is Where the Money Is" speech that changed the focus of security (1998), the presidency of Usenix Association (2000), the first call for the eclipse of authentication by accountability (2002), principal author of and spokesman for "Cyberinsecurity: The Cost of Monopoly" (2003), and cofounder of SecurityMetrics.Org (2004) and convener of Metricon 1.0 (2006).

© 2006 ACM 1542-7730/06/1100 \$5.00

MORE

Related articles in ACM's Digital Library:

Cerf, V. G. 2005. Spam, spim, and spit. *Communications of the ACM* 48(4).

Ford, R., Bush, M., Boulatov, A. 2005. Internet instability and disturbance: goal or menace? In *Proceedings of the 2005 Workshop on New Security Paradigms* (September).

Loibl, T. R. 2005. Identity theft, spyware and the law. In *Proceedings of the 2nd Annual Conference on Information Security Curriculum Development* (September).

McHugh, J. A. M., Deek, F. P. 2005. An incentive system for reducing malware attacks. *Communications of the ACM* 48(6).

Sukhai, N. B. 2004. Hacking and cybercrime. In *Proceedings of the 1st Annual Conference on Information Security Curriculum Development* (October).

These articles will be available online at www.acmqueue.com for an eight-week period beginning Nov. 1. Want full access? Join ACM today at www.acm.org and sign up for the Digital Library.