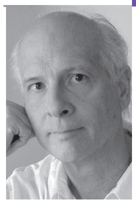
RIK FARROW

musings



rik@spirit.com

IT IS EASY TO BECOME A VICTIM OF future shock. I just read an ad in *New Scientist* for "gene silencers, suitable for *in vivo* work," by mail order. Once I had decided that the ad was real and not a joke, I next wondered whether any of my own genes deserved silencing via some mail-order sRNA sequences.

The ever-increasing scourge of Windows viruses, spyware, and rootkits provides another jolt of future shock. I've heard of people unplugging from the Internet rather than continue to deal with the plague of adware for porn sites, identity theft, and the requirement that they be clever enough to deploy at least two types of both anti-adware and antivirus software to be truly safe; some simply install some other OS, but I digress.

Sometimes Windows systems can become so infected with malware that the only way to secure them is to go through formatting and reinstallation. In *eWeek* (http://www.eweek.com/article2/0,1895,1945808,00.asp), an article quotes Mike Danseglio, program manager in the Security Solutions group at Microsoft, as saying that the only reliable solution is to rebuild from scratch. I'm pretty sure that most of you are not surprised to read this.

Securing Windows is not a simple problem. If it were, Microsoft would have laid this problem to rest years, and many billions of dollars, ago. Windows Vista, which makes some real progress in providing a more secure Windows environment by making it possible to use the system without being in the Administrator group and by running some device drivers unprivileged, will certainly help. But Vista has been delayed until at least January 2007. And even these changes will not address Windows' biggest issue, that of complexity. Real solutions to security issues will not be possible until Microsoft is willing to give up backward compatibility (see the December '05 opinion article by Dan Geer).

The Internet Is Broken

In a disturbing article in MIT's *Technology Review* (December 2005, http://www.technologyreview.com/InfoTech/wtr_16051,258,p1.html), David Talbot suggests that the "Internet is broken" and backs up this notion with support from David Clark, an early and key architect of the Internet.

Talbot writes that worms, spam, and phishing are evidence that the Internet needs replacing and that patching won't work. Besides confusing the Internet with end systems, Talbot does make some good points. The Internet was designed for just a few hundred systems, systems that were not mobile, and security was not even considered. Now, with the number of Internet-connected systems in the hundreds of millions, some of which are truly mobile systems (cell phones and PDAs are examples), the original Internet protocols seem a poor match with our current installed base.

Some of the architecture solutions suggested by Clark in Talbot's article make a lot of sense, whereas others just grate on my nerves. His first priority is giving "the medium a basic security architecture—the ability to authenticate who you are communicating [with] and prevent things like spam and viruses from ever reaching your PC." Whoa, there, Dr. Clark. Spam already comes from compromised systems, and certainly spam relay software will borrow the identity of the victim. Will we submit to iris scans in the future before we can send an email? And how in the world will a new Internet design defend vulnerable systems from exploitation?

There's more. The second point is to make the architecture "practical by devising protocols that allow Internet service providers to better route traffic and collaborate to offer advanced services without compromising their businesses." That part hints at creating a new, tiered Internet, one that permits ISPs to control traffic, giving those who pay for special services special access. Debate about the issue of content neutrality has arisen around the U.S. House bill known as the Barton Bill, after Representative Barton who wrote it, with Congress so far siding with neutrality. That is, large ISPs, such as AT&T, should not be able to filter out competing content, for example, paid music downloads from Google or Apple iTunes. And ISPs cannot add tariffs that make those offerings noncompetitive with the ISP's or their parent company's own offerings.

The whole idea of having telephone companies controlling the content their subscribers can receive strikes me as scary. Visions of *1984*, *V for Vendetta*, and good ol' Ma Bell running your communications media again just don't sit well with me. As if the telephone company has done a great job so far at controlling denial-of-service attacks and spam (e.g., those sales calls that used to occur at dinnertime), attacks (the random person who calls your phone number and starts screaming profanity at whoever answers, or the heavy breather who calls when your children are home alone). And then there's the telephonic version of phishing, where scammers call up elderly people and social-engineer them out of their savings. Sure, we trust the phone company to protect us and our ability to access information as we chose—just kidding.

But wait, Clark has two more points. I think the next one is actually very important: Allow future computing devices of any size to connect to the Internet. Right now, support for mobile IP, that is, the ability to maintain IP connections while you move from net to net, is extremely limited. Routing currently depends upon the network portion of your IP address, and if your device moves between networks, your IP address and your route must change. Changing your IP address plays havoc with protocols that embed the IP address in data, as well as killing any existing TCP connections. Most solutions focus on using a proxy that forwards your traffic to your current IP address, a clumsy solution that relies on some third party, the proxy service, as well as support for the applications you want to use, to work.

Mobile IP gets us right back into the territory of the telcos again. Imagine that we do somehow create a new Internet that supports real mobility.

Then, as you work, walk, or drive through cities with free WiFi, why use a costly cell phone, when you can use VoIP for free? There are already WiFi-enabled PDAs and cell phones, but not many. And most of these rely on Windows CE for their operating systems (what a scary idea). True, mobile IP will certainly impact telcos, but having this capability is really crucial to any new Internet design.

Finally, Clark suggests adding technology that makes the network easier to manage and more resilient. Like the third point, this is another strong argument for a reinvented Internet. I don't think Clark is talking about managing the Internet at subnet scales, but, rather, he is addressing the larger issues involved in managing the Internet, the network of networks. Back in the nineties, I would hear stories of how one large ISP would route its traffic over another ISP's network, preserving its own bandwidth, while taking advantage of a competitor. Today, these issues get resolved (more or less) through the careful configuration of BGP; still, they are not easy to solve. There are also issues such as slashdotting, DDoS, and other traffic-flow issues that really have no widely accepted management solution today.

Stupidity

Now, do I really believe that a new Internet will solve the security issues we see with today's Internet? Not at all. The real problems sit on people's desktops, and these involve insecure operating systems and applications. I believe that if it were possible to filter out all dangerous content, there would be a thriving market in doing so today. You have certainly observed that there is a huge market in selling incomplete and only partially effective solutions to viruses, spam, spyware, adware, rootkits, and other malware. I think you can compare the problem of blocking malware to the halting problem—in other words, it is an insolvable problem.

In the *eWeek* article, based on a presentation made at the InfoSec World Conference, Danseglio also said, "Phishing is a major problem because there really is no patch for human stupidity." Hmmm, we do let stupid people have bank accounts, right? They drive cars, pay taxes, raise children, own and use weapons; but we can't trust them to use their computers properly? Somehow I think this argument is specious. If using your computer results in damage to your bank account, is it your fault? Or is it the fault of the software that cannot parse email headers, validate domain names, or at least offer clear warnings such as "This Web site does not appear to be affiliated in any way with [fill in your financial institution here]." Or is it the fault of the underlying software that made it easy to install the spyware that stole your identity? Stupidity? If cars were as unsafe to use as today's computers, most people would still be walking.

When I learned how to fly small airplanes, I also learned that these same airplanes are designed in ways that make them safer than they might otherwise be. Stability is a big concern. Modern warplanes are inherently unstable, requiring clever fly-by-wire systems to make it possible for even well-trained pilots to fly them. Cessnas, in contrast, are designed so that they are stable, difficult to stall, and easy to land. Aircraft designers do this so that their product will be widely accepted and safer to use. Too bad our operating systems vendors haven't figured this out yet.

The Architecture Is Broken

I do believe that Microsoft, Sun, and the Open Source developers are working with a serious handicap. They are building and patching operating systems designed for hardware that is obsolete—hardware that was designed for another era. Our hardware architecture resembles that of ’60s mainframes, designed to support an operating system running a time-sharing system. We don’t run time-sharing systems anymore, and we haven’t for years. Most computers today have a single user, but the operating system designers have not come close to appreciating this fact. Remember that authors of the UNIX system quickly morphed the original, single-user version into a multiuser system, and every UNIX or Linux system today shares that legacy.

The single most dangerous and commonly exploited application today is the Web browser. Web browsers are purposely designed to execute remote code in the context of the single user of the system. No security system based on time-sharing notions, the Orange Book, Multi-Level Security (MLS), SELinux, or AppArmour is going to protect a user against code that that user has elected to execute. Today, reading HTML-formatted email and browsing the Web are the most insecure activities you can engage in. And the operating systems, and the hardware they rely upon, really don’t make the Web, and by extension the Internet, a very safe environment.

Time-sharing systems needed a method for isolating processes being executed by different users. Memory management does this and is itself controlled by software running at the highest privilege level, sometimes called ring 0. In today’s operating systems, all of the operating system—an enormous, complex program requiring megabytes of memory just for the code —runs at ring 0. A single error here compromises the entire system—and if this isn’t a bad way to design a system, I don’t know what is. But the hardware was designed for just such a system.

I would certainly like to think that the current environment is ripe for new designs and new ways of thinking about operating systems and security. But system architecture is not going to change easily, and neither are the operating systems that have been designed for these architectures.

The Lineup

But that’s enough bellyaching. In this issue of *;login:*, we start off with an opinion piece by Mark Burgess. Mark explains the meaning of autonomic computing, what it means today, and where it is going, in what I hope will initiate a series of articles about this topic.

In the Sysadmin section, Kurt Chan has satisfied my long quest for someone who can authoritatively explain the differences among different types of disk drives. I’ve heard people say that SCSI is dead and that SATA will supplant the more expensive SCSI drives. Chan explains that the problem with this analysis is that it doesn’t slice the problem correctly. While SCSI drives will be replaced by SAS (SCSI over Asynchronous Serial), the real divisions between drives have to do with how they will be used, not just the interfaces used to connect them. And no, SATA will not replace SAS. If you don’t believe me, just read Chan’s article.

Kirk McKusick has a different perspective about drive types, and he has contributed a short article that adds another way of looking at the drive types. Kirk sees the world from the filesystem and device-driver writer’s perspective, and this is relevant in its own way.

Next, Tom Haynes discusses the configuration and uses of ZFS, Sun’s Zone File System, coming to an Open Source system near you soon and already available in Solaris 10. Stefan Büttcher delves into a different filesystem aspect, indexing. Büttcher, whose paper about Wumpus was presented during last December’s FAST workshop, explains the design decisions behind

Wumpus, while explaining important issues about filesystem indexing on multiuser systems and systems using networked file systems.

In the security section, we start off with an article by Pablo Neira Ayuso, one of the key Netfilter developers. Neira explains the architecture supporting Netfilter's Connection Tracking subsystem, the foundation for stateful filtering in Linux kernels. Then Markos Gogoulos and Diomidis Spinellis report on their research into using live CDs for penetration testing.

This issue, as has become the custom, ends with articles by our regular columnists and book reviews.

I have, sadly, become accustomed to complaining about security. I recently wrote an article for a newsletter in which I pointed out that the proliferation of security vendors clearly demonstrates our collective failure to produce secure systems. Somehow, I don't think I will notice, or even believe, a little ad found in a science magazine that advertises that the solution to desktop security, and server security, can be obtained via mail order.

The solution would be a lot easier if only we were willing to stop using the software we rely upon today and start over. But Word has become the opium of computer users, and breaking the habit is not going to be easy. Perhaps a solution like ODF (Open Document Format) will be the methadone that eases us away from the addiction.