THE NEW REPUBLIC

What You Don't Know About Internet Security Will Definitely Hurt You

Jane Chong

October 22, 2013

This is the third installment in a series on whether and how to hold software makers financially liable for the insecurity of their products. Part I offered an overview of the problem of insecure code; Part II countered the notion that the technical challenges associated with minimizing software vulnerabilities weigh against the creation of any kind of maker-liability regime.

In the early twentieth century, typhoid struck mostly poor people. So it was odd when, in the summer of 1906, six of eleven people in the wealthy household of bank president Charles Henry Warren fell ill with the disease. Hired to investigate the source of the scourge, "sanitary engineer" George Soper followed the breadcrumbs to the Warren family's new, and recently-missing, cook. Piecing together her history, he learned that typhoid outbreaks had trailed Mary Mallon from house to house for a decade.

Mallon was a good cook with a tragic flaw: she did not wash her hands. This particular combination of talent and defect proved disastrous for her many patrons, since Mallon, later dubbed Typhoid Mary, was a rare, seemingly-healthy carrier of the fecal-oral bacterium *Salmonella typhi*. Eventually Mallon was apprehended, forcibly quarantined for three years and released on the condition that she would cease preparing food. But unconvinced that she was anything but hale, she took on a series of aliases and began to cook, and inadvertently kill, again.

To the modern reader, Mallon's denial of biology in the face of evidence is baffling—and criminal. But her conduct is less astonishing when translated to another context: the Internet.

Our collective behavior as Internet and software users is remarkably like Mallon's. End users are known to rely on easy-to-guess passwords, to unknowingly execute malware, and to neglect to timely install critical software patches. We do all this despite being told that these behaviors have costs, perhaps to ourselves, perhaps to others.

But if code creates real hazards for people and businesses, shouldn't that eventually generate a market for more secure code?

This is the simple and seductive argument of those who oppose liability for makers of insecure software: just leave the quality of code to the market to determine. The free market argument shows up commonly and often in greater detail to counter proposals for internet service provider

liability, but its logic operates similarly as opposition to holding software makers accountable for shipping vulnerability-ridden products. At its crux, this logic assumes that when it comes to society's cybersecurity needs, users can and should be the ones pulling the levers. Here is how Jim Harper of the libertarian Cato Institute put the point back in 2005: "On the margin, pushing disproportionate liability onto ISPs would erode Internet users' focus on self-awareness and self-help." Moreover, Harper noted, such a move would "suppress" what is "a well-developing and diverse market for Internet hygiene services."

In the software security context, this argument boils down to two parts: first, that patching practices and antivirus products have some handle on cybersecurity problems; and second, that shifting liability onto entities other than the user would interfere with the market's ability to generate its own remedies. It's an argument not unlike contending that in 1906 the market was equipped to handle the risk posed by Mallon—that rather than being quarantined, she should have been allowed to cook because New York families could develop good screening techniques for identifying infected food workers.

Security experts have written tomes on why monthly patch rollouts and steadily proliferating antivirus options do not collectively constitute a viable security solution to the problem of insecure code. But more can be said about the nature of this inadequacy, which traces back to the inadequacy of users. Consumers of "Internet hygiene services" are ultimately as ill-equipped to bear the burden of shaping the market to minimize software security risks as Mallon's employers were in controlling the spread of typhoid. The analogy applies on two levels, for as users we play the role of the victims—the New Yorkers who hired Typhoid Mary—but in important respects we also play the role of Mary herself.

Three features make Typhoid Mary a relevant analogy for the modern software user, and shed light on why relying on users to make responsible cyber hygiene decisions cannot make for a responsible national cybersecurity policy.

First, there is user apathy. The companies that produce buggy code are not alone in escaping the ramifications of their choices. Like Mallon, who remained healthy even as her patrons fell sick around her, users are not forced to suffer the full consequences of their personal use of buggy software or their bad security practices. This is a classic problem of what economists call negative externalities. And negative externalities are exacerbated by the fact that malware creators have gotten smarter about taking advantage of them.

Unlike the viruses and worms of yesteryear, which would typically disrupt the operation of the infected machine in a noticeable fashion, modern malware tends to secrete itself onto a machine and use the host to attack third parties. Experts estimate that 10 percent of U.S. computers have been infected and co-opted for remote exploitation by herders of sprawling, spam-spewing botnets. Botnets, increasingly the tool of choice for cybercriminals as a consequence of their inherent versatility, are made up of vast numbers of infected computers that, unbeknownst to their owners, operate in concert to distribute malicious code, disrupt Internet traffic or steal sensitive user data.

In 2010, Microsoft reported that more than 2.2. million PCs in the U.S. had been hijacked by bot herders. In June of this year, Microsoft's Digital Crimes Unit worked with the FBI and the U.S. Marshals Service to liberate more than 1,463 computers from the Citadel botnet, responsible for infecting an estimated 5 million computers worldwide and stealing $500 million from consumer and business bank accounts over 18 months.

Yet despite the continuing rise of botnets, many people lack reason to truly care that their computers are infected, because being part of a botnet does not especially harm them. In fact, people are on average quite unaware of how "pervasive and pernicious" the botnet threat is and remain unaware when their systems have been coopted.

This brings us to a second connection between Typhoid Mary and the computer user: ignorance. Users, commonly described as the weakest link in the security chain, generally lack the technical background to understand what is going on under the hoods of the various high-tech gadgets that make their worlds go round. So just as Mallon's ignorance as to the science of the spread of a pathogen made it all the easier for her to skip the soap and continue the cooking, our lack of understanding when it comes to the mechanics of cyber risks lends itself to poor cybersecurity hygiene, even as our reliance on the Internet—and our consequent risk—increases steadily.

Even the abstract knowledge that the internet is teeming with malicious activity does not seem to translate into an appropriate awareness of personal risk. In 2011, McAffee conducted a global study that showed that on average, consumers put their digital assets at a value of $37,438—and that more than a third of those consumers failed to institute protections across all those devices. Research conducted within other industries suggests that consumers tend to practice a kind of personal exceptionalism, believing they are less vulnerable to risks and less likely to be harmed by products than are others. As one security researcher points out, "[i]t stands to reason that any computer user has the preset belief that they are at less risk of a computer vulnerability than others." And here's the kicker: users do not necessarily exercise greater online discretion even when they have personally experienced an adverse event.

Technological illiteracy no doubt contributes to a litany of bad security practices. For example, in 2012, a Skype-commissioned survey of some 350,000 individuals revealed that 40 percent of adults do not update their software when prompted, and about a quarter skipped the updates because they did not understand the benefits. Users do not promptly patch software even when companies make such patches available in a timely fashion; something like 90 percent of successful exploits are successful attacks on unpatched systems. You might expect that users would be willing to timely deploy at least the most urgent fixes. But no, numerous studies have confirmed the widely held belief that users are extremely slow about deploying security fixes, even in the case of critical vulnerabilities. Indeed, the most infamous worms and viruses have exploited vulnerabilities for which patches were readily available. These include the Code Red worm in 2000, which caused an estimated $1.2 billion in network damage, and the SQL Slammer in 2003, an even faster-spreading worm that completely shut down the Internet in South Korea and led to outages and slowdowns throughout Asia.

Nothing better showcases the problems with dumping the burden of improving cybersecurity on the party with the least technical know-how to accomplish this than the emergence of one

distinct criminal enterprise: fake anti-virus software. The basic premise of the so-called "rogue" antivirus application is simple: feed on users' fear of malware to infect computers with malware. The scam sends an alert message to the user, offering a free (fake) scan and demanding a credit card number in exchange for removing the supposed infections. Researchers at Google recently conducted an analysis of 240 million web pages over 13 months and discovered that fake anti-virus software accounts for 15 percent of all malware on the web and for 50 percent of malware distributed through advertisements. The problem is growing, both in absolute terms and relative to other malware.

Put simply: we users are fools, and fools are easy to exploit.

There is a third factor keeping end-user liability from even approaching viability as a path toward better software security or better cybersecurity generally, and that's limited market power. Again, Typhoid Mary offers a useful analog. Under pressure to stop handling food, Mallon briefly attempted other occupations—taking a job, for example, as a laundress. Unfortunately, nothing paid as well for a woman of her station as did cooking. So cook she did.

Like Typhoid Mary, software users are hobbled by the limits of their market power. In an industry structured to reward fast shipping and eventual patching, software makers face no consequences for even knowingly shipping vulnerability-ridden products. Meanwhile, users lack the ability to determine the quality of proprietary software until it has become a standard. One commentator describes the nightmare simply: "The standardization process interacts with the unfortunate fact that latent software security defects tend to remain hidden until after software has become popular, and consequently, such defects play no role in the competition to set standards." Think about it. If you are dissatisfied with the security of your software, what are your options? Can you really afford to stop cooking altogether?

As a nation of modern-day Typhoid Marys, we pose a greater threat to the cyber ecosystem in which we operate than to ourselves. But unlike Mary, we cannot all be quarantined on an island next to Rikers. Our fate is just the opposite—to be increasingly interconnected, and increasingly exposed. So a smart cybersecurity policy has to be one that encourages cyber hygiene among users without mistaking it for an alternative to creating real demand for better security from software makers.

Software makers are distinctly unlike Typhoid Mary in that they have the knowledge and the capability to improve the security environment. They resemble Mallon in one respect, and one respect only: they lack adequate incentive to change their habits and have duly shunted the risks associated with bad code off on others. A nuanced software liability regime—one that holds software makers accountable for unacceptably flawed products as well as their negligent or reckless marketing—could correct this. It doesn't take a sanitary engineer to understand that.