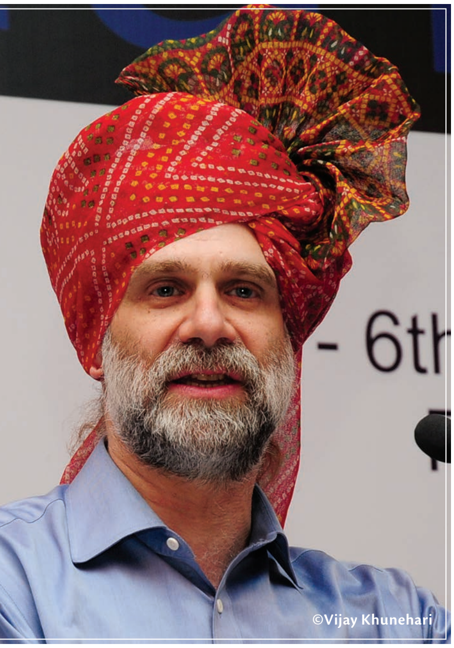

©Vijay Khunehari

Trust in Man/Machine Security Systems

Bruce Schneier
BT

I jacked a visitor's badge from the Eisenhower Executive Office Building in Washington, DC, last month. The badges are electronic; they're enabled when you check in at building security. You're supposed to wear it on a chain around your neck at all times and drop it through a slot when you leave.

I kept the badge. I used my body as a shield, and the chain made a satisfying noise when it hit bottom. The guard let me through the gate.

The person after me had problems, though. Some part of the system knew something was wrong, and wouldn't let her out. Eventually, the guard had to manually override something.

My point in telling this story is not to demonstrate how I beat the EEOB's security—I'm sure the badge was quickly deactivated and showed up in some missing-badge log next to my name—but to illustrate how security vulnerabilities can result from human/machine trust failures. Something went wrong between when I went through the gate and when the person after me did. The system knew it but couldn't adequately explain it to the guards. The guards knew it but didn't know the details. Because the failure occurred when the person after me tried to leave the building, they assumed she was the problem. And when they cleared her of wrongdoing, they blamed the system.

In any hybrid security system, the human portion needs to trust the machine portion. To do so, both must understand the expected behavior for every state—how the system can fail and what those failures look like. The machine must be able to communicate its state and have the capacity to alert the humans when an expected state transition doesn't happen as expected. Things will go wrong, either by accident or as the result of an attack, and the humans are going to need to troubleshoot the system in real time—that requires understanding on both parts. Each time things go wrong, and the machine portion doesn't communicate well, the human portion trusts it a little less.

This problem is not specific to security systems, but inducing this sort of confusion is a good way to attack systems. When the attackers understand the system—especially the machine part—better than the humans in the system do, they can create a failure to exploit. Many social engineering attacks fall into this category. Failures also happen the other way. We've all experienced trust without understanding, when the human part of the system defers to the machine, even though it makes no sense: "The computer is always right."

Humans and machines have different strengths. Humans are flexible and can do creative thinking in ways that machines cannot. But they're easily fooled. Machines are more rigid and can handle state changes and process flows much better than humans can. But they're bad at dealing with exceptions. If humans are to serve as security sensors, they need to understand what is being sensed. (That's why "if you see something, say something" fails so often.) If a machine automatically processes input, it needs to clearly flag anything unexpected.

The more machine security is automated, and the more the machine is expected to enforce security without human intervention, the greater the impact of a successful attack. If this sounds like an argument for interface simplicity, it is. The machine design will be necessarily more complicated: more resilience, more error handling, and more internal checking. But the human/computer communication needs to be clear and straightforward. That's the best way to give humans the trust and understanding they need in the machine part of any security system. ■

Bruce Schneier writes about security, technology, and people at www.schneier.com. His latest book is *Liars and Outliers: Enabling the Trust That Society Needs to Thrive* (Wiley, 2012). Schneier is starting a one-year fellowship at the Berkman Center for Internet and Society at Harvard University.

1540-7993/13/$31.00 © 2013 IEEE