[in]security

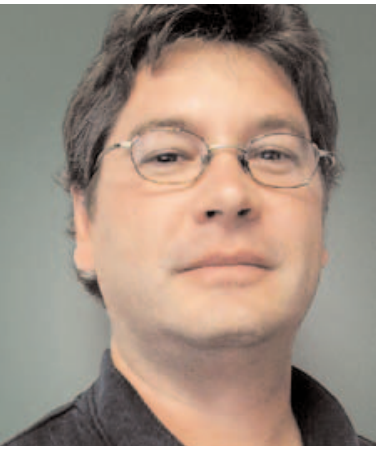
by Gary McGraw

Where does trust come from? : : :

Ken Thompson's famous 1984 Turing Award-winning paper, "Reflections on Trusting Trust," asks a simple question that turns out to be difficult to answer: When it comes to computers, what should we trust?

The Trusted Computing Group (TCG), an industry consortium whose members include Intel, Microsoft, AMD, and IBM, wants you to trust hardware—specifically, a coprocessor known as the Trusted Platform Module (TPM), which it claims can't be compromised by malicious software.

At first glance, the TPM may appeal to network security professionals, many of whom are understandably hesitant to trust software. To them, the pervasive nature of software exploits has undermined software's credibility. Yet these same people often trust the hardware that software runs on, arguing that hardware is somehow less prone to attack. Does this really make any sense?



TRUST AND TURTLES

Such thinking reminds me of a joke I once heard: A few years ago, a famous cosmologist who had just won a Nobel prize was asked to give a campus-wide address at the local university. The talk was very well-attended, with the audience including professors, students, and even a number of people from town. The cosmologist spoke of superclusters, galaxies, and the structure of solar systems.

When the talk was over, the cosmologist asked for questions from the audience. At that moment, a little old lady from the back of the lecture hall got up and made her way to the microphone. In a wavering voice, she said, "That was a very interesting talk, young man, but here in our neck of the woods we know that the Earth is actually held up by a turtle. Where does the turtle factor into your theories?"

Of course, the cosmologist was taken aback by such a crazy query, but after a quick bit of thought, he responded, "Thanks for the interesting question ma'am. Before I answer, I have a refining question for you. What holds up the turtle?"

The little old lady was stunned, thought for a moment, then responded, "You're a very clever young man, but it's turtles all the way down!"

WHO'S YOUR TURTLE?

The problem with trust in computer security is that it's turtles all the way down. Should you trust your software? How about your compiler? Did you write that OS? Did you burn in your own EEPROM? Did you fabricate your chips? Where does trust first get its footing?

In previous columns, we established that particularly advanced modern exploits are targeted at hardware, doing their dirty work by reprogramming the EEPROM, changing the BIOS, and burrowing below the OS. Though the hardware itself may not include the initial vulnerability leveraged to carry out a particular attack, it turns out to be at greater risk than we might anticipate if we simply trust it to be immutable and thus not prone to attack.

Given this view, hardware provides no more of a foundation for trust than software. Perhaps such hardware will leave us better off, but we need to reason through what we're trusting and why. In other words, we can't believe that hardware is secure simply because it's the bottom-most turtle.

> "Just as security is much more than a mechanism or a feature, trust can't be summed up in a coprocessor."

I believe that trust is earned, not given. Trust emerges as a byproduct of our interaction with people, organizations, and artifacts. Just as security is much more than a mechanism or a feature, trust can't be summed up in a coprocessor. If the main argument for adopting a security apparatus rests only on a basic misconception about trust, then we may falsely believe that we're getting more security than we actually are and thus trade away convenience, privacy, and openness in the name of security.

Gary McGraw is CTO of Cigital, a software quality management consultancy. He is co-author of *Exploiting Software* (Addison-Wesley, 2004), *Building Secure Software* (Addison-Wesley, 2001), and *Java Security* (Wiley, 1996). Reach him at gem@cigital.com.

: : :