We need handles for our knives and codes for our buildings

Carl Landwehr
George Washington University

A scenario that concerns me is one in which citizens have little idea and little technical basis on which to determine what software and systems they can depend on to maintain the security of their retail and banking transactions, the confidentiality of their medical records, and the safety of computer-controlled devices.

Unfortunately, this is not a "scenario", it's what we are experiencing today. Evidence is available in the daily newspapers. Last December we had the Target data theft, and in the past two weeks we've learned of the "Heartbleed" error that exposed untold volumes of data, including passwords, keys, and certificates, to theft over the past two years. Hopefully, the vulnerability was not actually exploited, but no one really knows, or can know, that it wasn't.

So to me the question is, how do we build a future in which this reality is improved upon? In particular, how can we build systems that provide some reasonable guarantees that the next release will at least provide some epsilon improvement in the assurance of security properties over the last one? Or even assurance that it won't actually degrade the security provided by the last one? This question will only become more critical with the proliferation of cyber physical systems of all sorts.

Here are two approaches that might help.

First, build software development tools that make it relatively easy to construct systems that, if not perfectly dependable for safety and security, at least can be certified to be free of commonly know exploitable types of vulnerabilities. As long at it is difficult for ordinary programmers to achieve this objective, as long as special training is required for programmers who have responsibilities for "secure coding," we are not going to have systems that are not loaded with latent vulnerabilities. We need to have tools that these programmers can use that will keep them from, so to speak, cutting their fingers on their own knives all the time.

Second, we need to organize both industry and the technical community to agree on what we know about how to build systems that can meet serious security requirements, and we need to codify that agreement into what, for lack of a better term, I have called a building code for building code. Today, we have Wyndham Hotels complaining in a legal brief that the FTC wants them to meet requirements for secure handling of data without having provided any guidance on what they are supposed to do to achieve that objective. While I suspect that Wyndham Hotels really wants the FTC to tell them what to do, it is perhaps correct to observe that security presently depends much too much on hiring a smart consultant to plug the holes in your systems rather than on assuring that the software delivered meets certain standards in the first place. The technical professions need to own up to the fact that security vulnerabilities are in fact engineering defects, not some kind of disease, and needs to start adopting measures that will reduce the number of such defects. Further, for a particular measure to be included in such a building code, there should be a sound basis for doing so. If such a basis can't be provided, then perhaps some research is needed to provide that basis.

There have been some remarkable advances in foundational security research, particularly in the area of cryptography, in the past few years. The feasibility of fully homomorphic encryption and the very recent results on obfuscation are exciting and may, within a decade or so, bear some practical fruit. We should pursue those avenues vigorously. But I think we should also see what we can do to make the programmer's job easier and to encourage industry to adopt less dangerous practices.