# Open Assurance

**Robin Bloomfield**
Associate Editor in Chief

During the past year, several high-profile reports of hacking and exfiltration of commercially confidential information have emerged. It is a reasonable assumption that some of our adversaries have access to the code and design details of the systems we deploy. Yet when it comes to the supply chain providing details of its products or regulators sharing why they think a critical system is trustworthy enough to be deployed, we are often in the dark. This can result in a strange information asymmetry between the adversary and those who have responsibility for a system's dependability. I think the supply chain should reflect on where its competitive advantage could really come from and reexamine what design and assurance information needs to be kept confidential in critical systems.

Perhaps when systems can kill humans, such as in embedded medical devices, do irredeemable damage to the environment, or destroy industry sectors or economies, we should have the right to know enough about these systems to form our own judgments about their trustworthiness and the benefits and risk trade-offs. It isn't just the code or design artifacts that we should have access to, but also their assurance—that is, all the reasons why these systems are trustworthy enough to be deployed.

But maybe we do not need such openness if we can trust regulators and their institutions in those areas without regulation, self-evaluation, and certification. Even if the present evaluation and regulatory approach works (the July/August 2013 issue of *IEEE Security & Privacy* that I guest-edited discusses medical device failures and the hundreds of deaths they appear to have caused), systems and supply chains are becoming more complex and lasting longer. In any case, many of the services we rely on daily aren't regulated. They're delivered by "accidental" systems of systems—those that have emerged in ways unanticipated by their designers.

So there is a multiplier effect here: threats are growing, assuring trustworthiness is getting harder. Because our systems are more open than we'd like, are the risks of open assurance—fully embracing openness of design and assurance—outweighed by the benefits?

The benefits of technically diverse and organizationally independent scrutiny are well-known and very hard to achieve, especially in monolithic organizations and cultures. Can we leverage others' perspectives through open source communities, crowdsourcing (there are even experiments in crowdsourcing highly technical aspects like formal verification; https://www.schafertmd.com/darpa/i2o/formalmethods/2013/july/pi/index.php?p=agenda_csfv), deployment of serious games, and competitions to get many minds engaging with the problems? Could recent failures such as the tsunami defenses at the Fukushima Daiichi nuclear plant, the financial bubbles of five years ago, or the licensing of airliners that catch fire have been identified and acted upon earlier if we had such an approach?

If we think about the workflow for crowdsourcing, four conditions must be met: a willing "crowd," modularity to enable many people to contribute incrementally, with increments small enough to be managed by an individual, and integration of assurance results at low cost and high quality.

To get the engagement of many, we would need assurance approaches and institutions that foster communities of interest; this would require social engineering and incentive structures that would make assurance interesting and engaging. We would need to maintain accountability by keeping responsibility with those who own and operate systems and avoid failures from assuming that everyone is evaluating something when in fact no one is.

To get modularity of effort, we need compositional approaches to design and assurance. While the concepts of layered assurance (www.acsac.org/2013/workshops/law) and compositional certification have been around for some time, in practice they are very hard to achieve. We need strong theories and models that allow us to make strong statements about system behavior and better tools and education that support effective

1540-7993/13/$31.00 © 2013 IEEE Copublished by the IEEE Computer and Reliability Societies

communication of highly technical arguments. This would also support the last condition for effective openness, the need for efficient integration of results. However, we are concerned not only with relatively closed problems—does the algorithm deadlock, does the code meet its specification—but also with how to deal with epistemic uncertainties, whether we have misjudged the real world and our assumptions are incorrect. These could include assumptions about technical artifacts (such as atomicity of instructions in a processor that fails under low-power fault conditions) or uncertainty in the wider world such as the height of tsunamis. Crowdsourcing seems to work in open problem areas of creative design; in situations when we don't know what we are looking for, but know when we see it, might it work here?

For some properties, modularity might not work, such as in the safety-critical domain. Statistical testing for reliability evaluation might be done on the system as a whole with complex models of the environment or plant. To leverage crowdsourcing for these properties, we might consider building APIs to large plant models to allow greater access, or we might develop methods of abstraction that allow us to efficiently simulate the complete system, at the expense of having to justify its fidelity. (We should avoid the trap of one recent account, during which the researcher confused the security of the simulator with the security of the actual avionics.)

So if all of the above succeed, we must have credible mechanisms for integrating the results: we must distinguish imaginative insights into why a system might fail from those that might only concern inconsequential noncompliance to standards. We would want to prevent too much review and chatter about unimportant findings, which leave experts swamped by questions and diverted from the real issues. A more formal basis to properties would help here, as would techniques for distinguishing between information and evidence.

In addition, if we can't update and evolve systems or their defenses, then learning more about their vulnerabilities can only help the attackers. So again we might need different system architectures and approaches to institutional learning than we have at present. Furthermore, in some systems—perhaps those of modest reliability—there can be safety and operational benefits of discovering and fixing vulnerabilities that might outweigh the security-related risks.

What is fascinating about this discussion is that the conditions we need for open assurance are really those for effective assurance. As with so many issues, it will be not be solved by a simple slogan, "open is good, closed is bad." We need principled, analytical approaches to achieving the balance between openness and confidentiality. Research by Peter Swire (http://ssrn.com/abstract=842228) provides a starting point for this, where he systematically considers the benefits of openness as a function of the effectiveness of the initial attack, the ability to alter the defense, the number of attacks, the learning by the attacker, and communication among attackers.

We must come to terms with the openness of networks, the vulnerability of systems, and the nature of threats they face and re-examine what design and assurance information needs to be kept confidential in critical systems. Does the empowerment enabled by the cloud, the globalization of Internet access, and education provide the resources so that more openness is not just an increased risk but a real benefit? Are we at a stage where we should make radical changes to how we assure software-based systems? ■