Trusting in Transparency

In providing security assurances, transparency and trust are inherently intertwined concepts, but their relationship is not well understood.

The slightly tarnished image of the computer industry, in terms of its ability to maintain security and privacy for business and private users, has played a role in impeding the growth and acceptance of some e-commerce and e-government products and services in recent years. One reason for this mistrust may be the perception (or actuality) that hosts and providers have lost control of the digital data transport medium as well as the software infrastructure that supports it.

The fact remains that computers (on or off of the Internet), for the foreseeable future, will be increasingly subjected to a variety of aggressive attacks for which none but palliative or patchwork solutions have been presented. Consumers have grown increasingly skeptical of (or annoyed by) lock icons, digital signatures, passwords, privacy policy statements, and other techniques now commonly used to provide security assurances. Perhaps this is because none of this addresses the real problem, which is that consumers have no obvious way of determining how to trust the systems they choose or are required to use.

What is Trust?

At the root of this situation lie general questions about the nature of trust, a utopian ideal that is not well defined. This makes it problematic to attempt to implement or integrate vague notions about trust into computational or rule-based systems.

Many varied meanings of the word trust are invoked when computing professionals use the phrase "trusted computing." The overriding definition is that of reliance or dependence, as in "I trust that you will …" but there is also some optimism or hope in the future, as with "we can trust that the outcome should …" Certainly the concept of custody or care is also involved, with "placed in the trust of …" emphasized in commercial or social entity relationships, such as charitable trusts, land trusts, bank trusts, and living trusts.

But many trusting aspects of human nature can be exploited through cons like Ponzi schemes and other affinity fraud tactics—including the ways that ratings on Amazon and eBay have been manipulated to indicate unjustified trust levels.

As well, choosing to trust often involves a mystical transcendence—the phrases "blind trust" and "absolute trust" come to mind. When situations are too complex for humans to comprehend, some find salvation in the idea of placing trust in a deity. A similar type of "all-knowing, all-powerful" transfer of trust is

MICHAEL KLOSE

Since computers are inherently somewhat opaque, one must ascertain whether the level of transparency provided is sufficient to ensure trust in the system.

increasingly yielded to computers, even when evidence of reliability or safety is sorely lacking. The use of the terms "artificial intelligence" and "expert systems" further enhances the sense of trust in computers that are no more knowledgeable, and often less so, than the fallible humans who designed them.

One interesting view of trust involves the interactive effects of intense, shared experiences and travail on its development. Obvious examples are found in reality TV shows and corporate team-building events, where participants leave with a feeling (albeit sometimes fleeting) of increased trust and bonding. Helen Nissenbaum [7] indicates that to the extent security hurdles (if not overly cumbersome) sometimes impede ease of use, these may have the inadvertent side effect of increasing travail, so high usability may not necessarily be the best goal.

Closed vs. Open Source

When trust is questionable, candor often plays a role in providing assurances. The idea that transparency or openness increases trust is embodied in the word "antitrust"—the creation of increased confidence through the process of breaking up business trusts that might privately engage in practices that discourage marketplace competition. Often in governments, competitive interests are served and trust is enhanced through policies that reveal hidden agendas, such as sunshine laws and the U.S. Freedom of Information Act. Similarly, the opposing concepts of "trust me" versus "trust yourselves" along with transparency, or lack thereof, are certainly evident in the various camps of software developers.

At one extreme, we have the legacy view of security by obscurity [6], a philosophy that maintains that by concealing source code and design, one can prevent or minimize malicious activity. The ongoing proliferation of malware (such as the Blaster worm that deposited backdoor Trojan horse software for later use, which, over several days, infected more than a half-million Windows-based computers worldwide), is proof positive that the closed source technique is far from secure.

The opposite side is characterized by the open source movement, where community review is viewed as a way of ferreting out and correcting software flaws that might otherwise have been exploited. Although open source supporters claim their products appear to suffer fewer problems, this may be attributed more to their smaller percentage of the marketplace than to an inherently rugged nature, as the number of Linux worms and other open source attacks continues to grow.

As it turns out, neither closed nor open source code examinations can provide total assurance of program correctness, because of the computational complexity issues that make it infeasible to determine that computer software will perform only the tasks it was designed for, and no more. Quite simply, it is impossible to differentiate the code you want running in a program from the code you don't want, on a generic basis.

Auditability

Yet, despite ongoing concerns about integrity and security, we trust large quantities of critical data and processes to computational systems. Since decision making often relies on collections of data that must be accurate and reliable, additional confidence is typically provided through redundancy and auditability. The double-entry model for accounting is illustrative of this methodology—separate sets of books are independently maintained and then cross-checked for accuracy by auditors who are deemed competent and trustworthy. Layers of assurance are therefore created by

the systematic process used, with the expectation that flaws will be exposed and mitigated. Clearly, this does not always succeed, as evidenced by the rash of fraudulent accounting practices uncovered in recent years.

The problem arises when internally conflicting goals (such as profit incentives) reduce the amount of transparency necessary in order to apply the auditing process appropriately. Since computers, like people, are inherently somewhat opaque, one must ascertain whether the level of transparency provided is sufficient to ensure trust in the system.

This balance between transparency and trust can be considered in terms of a concentration or broadening of vulnerability, as cyber-journalist Lynn Landes has observed in an "eggs in one basket" theory of risk distribution (see www.ecotalk.org). With security by obscurity, transparency is deemed inversely proportional to trust, and the risk is focused on a few (or perhaps many more, as with Microsoft) employees. With open source, transparency is roughly equivalent to trust (or at least it provides a great deal of it), and risk is spread globally.

Certification

Somewhere in the middle, between open and closed source, we have certification programs such as the International Standards Organization's Common Criteria (see csrc.nist.gov/cc/) and the National Institute of Standards and Technology's Digital Signature and Secure Hash (see www.itl.nist.gov/fipspubs/fip180-1.htm) that provide imprimaturs used to ascribe confidence in the security of software products, the processes used to develop them, and their correct embodiment in distributed units.

Mandated by U.S. Congress with the Computer Security Act of 1987 (Public Law 100-235), is the U.S. Department of Defense's Trusted Computer System Evaluation Criteria, known as TCSEC (see www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.pdf). TCSEC was originally intended for application to "sensitive information" whose "loss, misuse or unauthorized access to or modification of which could adversely affect the national interest or the conduct of Federal programs, or the privacy to which individuals are entitled." This methodology, and its successor, the Common Criteria (administered by NIST's Computer Security Resource Center), have also been adopted voluntarily in other settings, such as health care and banking.

As with many government projects, though, the devil is in the details. In this case, the bottom-up approach used in TCSEC (that of creating security levels, somewhat analogous to security clearances for personnel, associated with features that mitigate risks) was too inflexible to accommodate modular and object-oriented designs. TCSEC's step-wise gradations meant that the security bar could be globally set too high, or not high enough, for an entire system, in an effort to balance protection from worst-case scenarios against overly compromised realizability.

Conversely, the Common Criteria evolved as a top-down or "Chinese menu" style of providing assurances through the process of identification of component features and their associated risks. But this piecewise method, by allowing the vendor or purchaser to specify the protections to be implemented, suffers from the problems of unintended and overlooked consequences. For example, although the Common Criteria deals adequately with numerous interdependencies (such as, if you implement X, then you must implement Y and perhaps also Z), it fails to include any mappings for counterindications (if you implement J then you cannot also implement K or L) [5].

Ultimately, any metric for evaluation using a trusted computing approach is tied to the understanding, intuition, and honesty of the system designers and evaluators in providing an appropriate selection of assurance components. Unfortunately, with the Common Criteria, often the resulting Protection Profile is necessarily so complex and detailed that (other than perhaps as a checklist for certification or procurement) reliance on it for comprehensive security guarantees is unlikely, except for the most simplistic of systems. This is not to

That transparency might need to be managed in order to remain competitive is a concept that strikes a discord with legacy firms used to doing business the old-fashioned way, behind closed doors.

say that the program is useless, just that it has shortcomings, and many of those stem from obfuscation through the imposition of a nest of details. Hence, it suffers from a lack of transparency, which is ironic, since the exercise of elaborating the details was obviously intended to provide such.

Full Disclosure

That transparency might need to be managed in order to remain competitive is a concept that strikes a discord with legacy firms used to doing business the old-fashioned way, behind closed doors. But trust and transparency are not necessarily synonymous with full exposure, as Don Tapscott and David Ticoll explain in their book *The Naked Corporation* [8].

Rather than appear as an emperor without clothes, protected by only the gossamer of public relations spin, some organizations are instead choosing to develop an "open kimono" culture by proactively engaging in transparency assessments and adjustments. As Tapscott said, "if corporations are going to be naked, they'd better be buff." His comments stress that "undressing for success" cannot merely be a veneer, but rather it requires abiding by basic values in all operations—telling the truth, honoring commitments, considering stakeholder interests, being candid about shortcomings, and building and delivering the best products. This is good advice for day-to-day operations as well as disclosures involving security matters.

Cryptography

With e-cash, e-voting, and other transactional applications, cryptographic approaches have been considered to enhance security. Indeed, David Chaum has claimed it should be possible to provide comprehensive assurances via cryptographic techniques that are independently verifiable [1]. He has even devised an ingenious voting scheme, using overlays to display ballot choices and mix-nets to maintain anonymity, but it should be noted this method necessarily includes a way in which voters can confirm their selections on a tangible and human-readable medium.

But cryptography itself poses a host of transparency problems. First, there are the cryptographic keys that must be distributed securely and maintained in such fashion that disallows collusion among administrators (referred to, of course, as "trustees") while preventing interception. Then there is the algorithm itself, which must be subjected to a thorough correctness proof, even if it appears to be verifiable (such as through the use of homomorphism). The issue of obsolescence of the algorithm, due to increased computational speeds and novel cracking approaches must continually be assessed. Finally, the implementation of the algorithm in software and/or hardware must also be demonstrated to be correct through rigorous, end-to-end provability. Under the Common Criteria program, satisfaction of these constraints would require certification at Evaluation Assurance Level 7, which no product has yet attained.

Furthermore, it is likely that none of this assessment process will be comprehensible by average (or even somewhat above average) end users and administrators, so additional transparent assurances must be provided to indicate that the embodiment has not been altered from the approved version. It is therefore imperative to consider and mitigate the impact of this lack of transparency on the human trust of the systems in which cryptographic processes are deployed. Most users are now savvy or skeptical enough to know that simple statements like "we use cryptography to secure your data" are no longer acceptable consolations, especially if problems arise.

The transparency issues of cryptography aside, another one of its shortcomings is that although the parcel of information it encodes may be secure during data transmission, the ends (before and after encryption) may not be, and these are still vulnerable to attack. Conversely (whether cryptography is or is not present), users may be led to believe that the middle of a process is secure, as when they are unwittingly using a spoofed or compromised Web site to enter data. This problem with rampant theft of credit card information in online transactions led Visa to implement its cardholder information security program and MasterCard to create a site data protection standard. Smaller outlets that do not have the capability to secure their own Web site can contract with third-party services that can act as filters between customers and credit agencies.

Although this middle-man approach may reassure creditors and corporations somewhat, it is only a stop-gap technique that does not address the overriding problems of online fraud and identity theft. Since anti-fraud heuristics are usually applied privately, it may not be possible to assess the amount of false positives and the impacts on fair access to services and user privacy. If the trust needed for e-commerce solutions is increased by sacrificing trust in the customers, this inverse relationship will inevitably become too adversarial to survive. Consumers who are subjected to absurd identity checks (such as reciting the three-digit number on the back of credit cards) like stooges in a security shell game, will use other providers that are viewed as less antagonistic. Other solutions must be sought.

Trust and Risk

Perhaps security folk have been addressing this problem from the wrong direction, by first assessing risk and then devising controls to mitigate it in order to earn trust. What we may instead need to do is assess trust, and then determine appropriate levels of risk. The topsy-turvy view is that trust enables risk taking. Economists and sociologists refer to this as "social capital" [2].

One way this type of capital can be provided is with threshold cryptography, where risk is managed by distributing trust to multiple parties. The Secure Electronic Transaction (SET) system used by MasterCard and Visa applies this method in order to force adversaries to take more risks by penetrating numerous systems rather than just one [3]. Of course, layering and middle-man approaches also increase complexity, which necessarily reduces transparency, so there may be diminishing returns to this technique. Nevertheless, the "view from the outside" philosophy differs significantly from the traditional risk-based and asset-protection methods (as I had described in my first column in this series [4]) such that it might be fruitful to consider its relevance within new paradigms for control structures and assurance evaluations. This, along with methods for parameterizing and transferring levels of trust, are subjects for further elaboration.

Conclusion

Transparency is playing an increasingly important role in the world of computer security. But as with many sociological interactions with technology, an optimal balance is difficult to quantify. The consideration of a trust-centric approach (as opposed to a vulnerability-based one) may help achieve the transparency needed to ensure confidence and reduce perceived (and perhaps even actual) risks in transactional experiences. C

References

1. Chaum, D. Secret ballot receipts and transparent integrity, May 2002; grouper.ieee.org/groups/scc38/1583/meeting_docs_-_020514/.
2. Farrell, C. Sound money. June 8, 2002; soundmoney.publicradio.org.
3. Frankel, Y. and Yung, M. Risk management using threshold RSA cryptosystems. *Usenix*, 1998.
4. Mercuri, R.T. Computer security: Quality rather than quantity. *Commun. ACM 45*, 10 (Oct. 2002).
5. Mercuri, R.T. Uncommon criteria. *Commun. ACM 45*, 1 (Jan. 2002).
6. Neumann, P.G. and Mercuri, R.T. Security by obscurity. *Commun. ACM 46*, 11 (Nov. 2003).
7. Nissenbaum, H. Securing trust online: Wisdom or oxymoron? *Computer Ethics: Philosophical Enquiry*. New York University School of Law, 2000.
8. Stepanek, M. Don Tapscott on transparency. *CIO Insight* (Oct. 2003).

Rebecca T. Mercuri (mercuri@acm.org) is a fellow at Harvard University's Radcliffe Institute for Advanced Study.