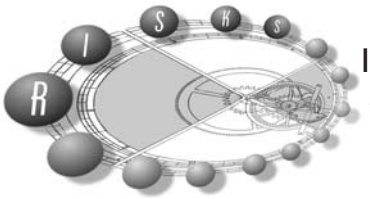

Uncommon Criteria

The software development process can benefit from the use of established standards and procedures to assess compliance with specified objectives, and reduce the risk of undesired behaviors. One such international standard for information security evaluation is the Common Criteria (CC, ISO IS 15408, csrc.nist.gov/cc). Although use of the CC is currently mandated in the U.S. for government equipment (typically military-related) that processes sensitive information whose "loss, misuse, or unauthorized access to or modification could adversely affect the national interest or the conduct of Federal programs" (Congressional Computer Security Act of 1987), it has been voluntarily applied in other settings (such as health care). In the U.S., oversight of CC product certification is provided by the National Institute of Standards and Technologies (NIST).

The goal of the CC is to provide security assurances via anticipation and elimination of vulnerabilities in the requirements, construction, and operation of IT products through testing, design review, and implementation. Assurance is expressed by degrees, as defined by selection of one of seven Evaluation Assurance Levels (EALs), and derived by assessment of correct implementation of the security functions appropriate to the level selected, and evaluation in order to obtain confidence in their effectiveness.

However, the use of standards is not a panacea because product specifications may contain simultaneously unresolvable requirements. Even the CC, regarded as a state-of-the-art standard, disclaims its own comprehensiveness, saying it is "not meant to be a definitive answer to all the problems of IT security. Rather, the CC offers a set of well-understood security functional requirements that can be used to create trusted products or systems reflecting the needs of the market." The CC methodology falls short in addressing and detecting potential design conflicts.

This major flaw of the CC is directly related to its security functional requirement hierarchy. In selecting an EAL appropriate to the product under evaluation, the CC specifies numerous dependencies among the items necessary for implementing a level's criteria of assurance. In essence, it formulates a mapping whereby if you implement X, you are required to implement Y (and perhaps also Z, and so on). But the CC fails to include a similar mapping for counterindications, and does not show that if you implement J then you cannot implement K (and perhaps also not L, and so on).

A good example of how this becomes problematic arises when both anonymity and auditability are required. The archetypical application of such simultaneous needs occurs in off-site election balloting, but one can also find this in such arenas as Swiss-style banking or AIDS test reporting. If the CC process were used with voting (no standards have been mandated, but NIST involvement is now being considered), it must assure that each ballot is cast anonymously, unlinkably, and unobservably, protecting the voter's identity from association with the voting selections. Because access to the ballot-casting modules requires prior authentication and authorization, pseudonymity through the use of issued passcodes provides a plausible solution. But the CC does not indicate how it is possible to maintain privacy while also resolving the additional requirement that all aliases must be traceable back to the individual voters in order to assure validity.

Furthermore, the need for anonymity precludes the use of traditional transaction logging methods for providing access assurances. Randomized audit logs have been proposed by some voting system vendors, but equipment or software malfunction, errors, or corruption can easily render these self-generated trails useless. Multiple electronic backups provide no additional assurances; if the error occurs between the point of user data entry and the writing of the cast ballot, all trails would contain the same erroneous information. Pure anonymity and unlinkability, then, are possible only if authentication and authorization transactions occur separately from balloting, but this is difficult to achieve in a fully electronic implementation.

The remedy to this and other such flaws in the CC involves augmentation with extensions that go beyond the current standard. For voting, one solution is to produce voter-verified paper ballots for use in recounts. Thus, the use of the CC in the secure product development cycle is encouraged, but prudent application and consideration of risks imposed by conflicting requirements is also necessary. C

MARTIN MAYO

Rebecca Mercuri (mercuri@acm.org) is an assistant professor of CS at Bryn Mawr College; www.notablesoftware.com/evote.html.