Chapter 2.20 Identifying Security Requirements Using the Security Quality Requirements Engineering (SQUARE) Method

N. R. Mead
Carnegie Mellon University, USA

ABSTRACT

In this chapter, we describe general issues in developing security requirements, methods that have been useful, and a method (SQUARE) that can be used for eliciting, analyzing, and documenting security requirements for software systems. SQUARE, which was developed by the CERT Program at Carnegie Mellon University's Software Engineering Institute, provides a systematic approach to security requirements engineering. SQUARE has been used on a number of client projects by Carnegie Mellon student teams, prototype tools have been developed, and research is ongoing to improve this promising method.

THE IMPORTANCE OF REQUIREMENTS ENGINEERING

It is well recognized in industry that requirements engineering is critical to the success of any major development project. Several authoritative studies have shown that requirements engineering defects cost 10 to 200 times as much to correct once fielded than if they were detected during requirements development. Other studies have shown that reworking requirements defects on most software development projects costs 40 to 50% of total project effort, and the percentage of defects originating during requirements engineering is estimated at more than 50%. The total percentage of project budget due to requirements defects is 25 to 40%.

A recent study found that the return on investment when security analysis and secure engineering practices are introduced early in the development cycle ranges from 12 to 21%, with the highest rate of return occurring when the analysis is performed during application design (Soo Hoo, Sudbury, & Jaquith, 2001). The National Institute of Standards and Technology (NIST) reports that software that is faulty in security and reliability costs the economy $59.5 billion annually in breakdowns and repairs (NIST, 2002). The costs of poor security requirements show that even a small improvement in this area would provide a high value. By the time that an application is fielded and in its operational environment, it is very difficult and expensive to significantly improve its security.

Requirements problems are the number one cause of why projects:

- Are significantly over budget
- Are significantly past schedule
- Have significantly reduced scope
- Deliver poor-quality applications
- Are not significantly used once delivered
- Are cancelled

Requirements engineering typically suffers from the following major problems:

- Requirements identification typically does not include all relevant stakeholders and does not use the most modern or efficient techniques.
- Requirements analysis typically is either not performed at all (identified requirements are directly specified without any analysis or modeling) or analysis is restricted to functional requirements, ignoring quality requirements, other non-functional requirements, and architecture, design, implementation, and testing constraints.
- Requirements specification is typically haphazard, with specified requirements being ambiguous, incomplete (e.g., non-functional requirements are often missing), inconsistent, not cohesive, infeasible, obsolete, neither testable nor capable of being validated, and not usable by all of their intended audiences.
- Requirements management is typically weak with poor storage (e.g., in one or more documents rather than in a database or tool) and missing attributes, and is limited to tracing, scheduling, and prioritization. (http://www.sei.cmu.edu/cmmi/)

Security Requirements Issues

Although security requirements are often identified during the system life cycle, they tend to be general mechanisms such as password protection, firewalls, virus detection tools, and the like. Often the security requirements are developed independently of the rest of the requirements engineering activity and hence are not integrated into the mainstream of the requirements activities. As a result, security requirements that are specific to the system and that provide for protection of essential services and assets are often neglected.

In typical requirements documents, security requirements, when they exist, are in a section by themselves and have been copied from a generic set of security requirements. The requirements elicitation and analysis that is needed to get a better set of security requirements seldom takes place.

Much requirements engineering research and practice has addressed the capabilities that the system will provide. So a lot of attention is given to the functionality of the system, from the user's perspective, but little attention is given to what the system should *not* do. In one discussion on requirements prioritization for a specific large system, ease of use was assigned a higher priority than security requirements. Security requirements were in the lower half of the prioritized requirements. This occurred in part because the only