SEI SERIES • A CERT® BOOK

SOFTWARE SECURITY SERIES

Software Security Engineering

A Guide for Project Managers

Julia H. Allen • Sean Barnum
Robert J. Ellison • Gary McGraw
Nancy R. Mead

Software Security Engineering

Misuse/Abuse Case Templates

Templates for misuse and abuse cases appear in a number of references. They can be text or diagrams, and some are supported by tools. Some good sources for templates are in materials by Sindre and Opdahl [Sindre 2001] and Alexander [Alexander 2002]. Figure 3–1 is an example of a use/misuse-case diagram to elicit security requirements from Alexander's article. The high-level case is shown on the left; use cases are drawn in white and misuse cases are drawn in black.

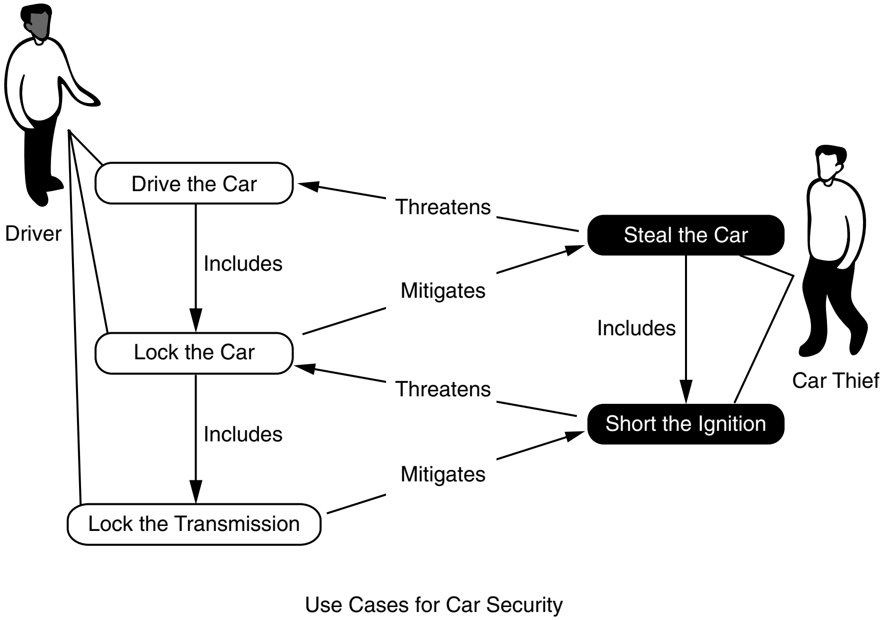

Figure 3–1: *Misuse case example*

some guidance in this regard (see Section 2.3.2). Attack patterns are akin to patterns in sewing—that is, a blueprint for creating an attack. Everyone's favorite example, the buffer overflow, follows several different standard patterns, but patterns allow for a fair amount of variation on a theme. They can take into account many dimensions, including timing, resources required, techniques, and so forth [Hoglund 2004]. When we're trying to develop misuse and abuse cases, attack patterns can help.

It is possible for misuse cases to be overused (and generated forever with little impact on actual security). A solid approach to building them requires a combination of security know-how and subject matter expertise to prioritize misuse cases as they are generated and to strike the right balance between cost and value.

Although misuse and abuse cases can be used as a stand-alone activity, they are more effective when they are developed as part of an overall security requirements engineering process. As noted in Section 3.1.3, a number of processes can be used to address security requirements engineering. In the next section, we describe one such process, the SQUARE process model, in which misuse and abuse cases play important roles. Consult the reference material that we have provided to learn about other processes and select the process and methods that are best for your organization.

3.3 The SQUARE Process Model M L L3

Security Quality Requirements Engineering (SQUARE) is a process model that was developed at Carnegie Mellon University [Mead 2005].[3] It provides a means for eliciting, categorizing, and prioritizing security requirements for information technology systems and applications. (Note that this section and the following sections all discuss security requirements, regardless of whether the term "security" is specifically used as a qualifier.) The focus of the model is to build security concepts into the early stages of the SDLC. It can also be used for documenting and analyzing the security aspects of systems once they are implemented in the field and for steering future improvements and modifications to those systems.

After its initial development, SQUARE was applied in a series of client case studies [Chen 2004; Gordon 2005; Xie 2004]. Prototype tools were also developed to support the process. The draft process was revised and established as a baseline after the case studies were completed; the baselined process is shown in Table 3–1. In principle, Steps 1–4 are actually activities that precede security requirements engineering but are necessary to ensure that it is successful. Brief descriptions of each step follow; a detailed discussion of the method can be found in [Mead 2005].

[3] The SQUARE work is supported by the Army Research Office through grant number DAAD19-02-1-0389 ("Perpetually Available and Secure Information Systems") to Carnegie Mellon University's CyLab.

Table 3–1: *The SQUARE Process*

Number	*Step*	*Input*	*Techniques*	*Participants*	*Output*
1	Agree on definitions	Candidate definitions from IEEE and other standards	Structured interviews, focus group	Stakeholders, requirements engineers	Agreed-to definitions
2	Identify security goals	Definitions, candidate goals, business drivers, policies and procedures, examples	Facilitated work session, surveys, interviews	Stakeholders, requirements engineers	Goals
3	Develop artifacts to support security requirements definition	Potential artifacts (e.g., scenarios, misuse cases, templates, forms)	Work session	Requirements engineers	Needed artifacts: scenarios, misuse cases, models, templates, forms
4	Perform (security) risk assessment	Misuse cases, scenarios, security goals	Risk assessment method, analysis of anticipated risk against organizational risk tolerance, including threat analysis	Requirements engineers, risk expert, stakeholders	Risk assessment results

Continues

Table 3–1: *The SQUARE Process (Continued)*

Number	*Step*	*Input*	*Techniques*	*Participants*	*Output*
5	Select elicitation techniques	Goals, definitions, candidate techniques, expertise of stakeholders, organizational style, culture, level of security needed, cost–benefit analysis	Work session	Requirements engineers	Selected elicitation techniques
6	Elicit security requirements	Artifacts, risk assessment results, selected techniques	Accelerated Requirements Method, Joint Application Development, interviews, surveys, model-based analysis, checklists, lists of reusable requirements types, document reviews	Stakeholders facilitated by requirements engineers	Initial cut at security requirements

Table 3–1: *The SQUARE Process (Continued)*

Number	*Step*	*Input*	*Techniques*	*Participants*	*Output*
7	Categorize requirements as to level (e.g., system, software) and whether they are requirements or other kinds of constraints	Initial requirements, architecture	Work session using a standard set of categories	Requirements engineers, other specialists as needed	Categorized requirements
8	Prioritize requirements	Categorized requirements and risk assessment results	Prioritization methods such as Analytical Hierarchy Process (AHP), triage, and win-win	Stakeholders facilitated by requirements engineers	Prioritized requirements
9	Inspect requirements	Prioritized requirements, candidate formal inspection technique	Inspection method such as Fagan and peer reviews	Inspection team	Initial selected requirements, documentation of decision-making process and rationale

3.3.1 A Brief Description of SQUARE

The SQUARE process is best applied by the project's requirements engineers and security experts in the context of supportive executive management and stakeholders. We have observed that this process works best when elicitation occurs after risk assessment (Step 4) has been done and when security requirements are specified before critical architecture and design decisions. Thus critical security risks to the business will be considered in the development of the security requirements.

Step 1, "Agree on definitions," is needed as a prerequisite to security requirements engineering. On a given project, team members tend to have definitions in mind, based on their prior experience, but those definitions often differ [Woody 2005]. For example, for some government organizations, security has to do with access based on security clearance levels, whereas for others security may have to do with physical security or cybersecurity. It is not necessary to invent definitions. Sources such as the Institute for Electrical and Electronics Engineers (IEEE) and the Software Engineering Body of Knowledge (SWEBOK) provide a range of definitions to select from or tailor. A focus group meeting with the interested parties will most likely enable the selection of a consistent set of definitions for the security requirements activity.

Step 2, "Identify security goals," should be done at the organizational level and is needed to support software development in the project at hand. This step provides a consistency check with the organization's policies and operational security environment. Different stakeholders usually have different goals. For example, a stakeholder in human resources may be concerned about maintaining the confidentiality of personnel records, whereas a stakeholder in a financial area may be concerned with ensuring that financial data is not accessed or modified without authorization. It is important to have a representative set of stakeholders, including those with operational expertise. Once the goals of the various stakeholders have been identified, they need to be prioritized. In the absence of consensus, an executive decision may be needed to prioritize them. It is expected that the goals identified in this step will link to the core properties discussed in Chapter 2.

Step 3, "Develop artifacts," is necessary to support all subsequent security requirements engineering activities. Organizations often do not have a documented concept of operations for a project, succinctly