Modeling Security Requirements for Trustworthy Systems

Kassem Saleh
American University of Sharjah, UAE

Ghanem Elshahry
American University of Sharjah, UAE

BACKGROUND

To increase users' trust in the systems they use, there is a need to develop trustworthy systems. These systems must meet the needs of the system's stakeholders with respect to security, privacy, reliability, and business integrity (Mundy, deVries, Haynes, & Corwine, 2002). The first major step in achieving trustworthiness is to properly and faithfully capture the stakeholders requirements. A requirement is something that the system must satisfy or a quality that the system must possess. A requirement is normally elicited from the system stakeholders, including its users, developers, and owners. Requirements should be specified before attempting to construct the system. If the correct requirements are not captured properly and faithfully, the correct system cannot be built. Consequently, the system will not be usable by its intended users. The success of any system depends on meeting requirements classified under two complementary types. First, the functional requirements are the system's operations from the user's perspective describing the visible and external interactions with the system under consideration. Second, the non-functional requirements (NFRs) are mainly the system's constraints imposing special conditions and qualities on the system to construct. Consequently, system acceptance testing must be based on both functional and non-functional system's requirements. Unfortunately, it is reported that about 60% of errors originate from the requirements and analysis activities (Weinberg, 1997).

Surveys have shown that large numbers of IT-based systems were implemented starting from their elicited functional requirements without a clear and formal consideration of their non-functional counterparts such as security requirements. Furthermore, system requirements engineers and analysts are not well-trained in capturing security requirements early in the system development process. Security assurances are often based on the traditional and ad hoc approach of conducting penetration tests followed by a patching process. This approach is very costly and endangers the fulfillment of the basic goals of system security, namely confidentiality, integrity, availability, and accountability. Recently, many researchers addressed security requirements engineering as an integral and essential element of systems engineering. Devanbu and Stubblebine (2000) propose a roadmap for software engineering for security, and Henning and Garner (1999) consider life cycle models for survivable and secure systems.

Non-functional requirements can be classified under three broad categories (Robertson & Robertson, 1999): system-related, process and project-related and human-related requirements.

The rest of this article is organized as follows. The next section overviews the security goals and requirements. The third section introduces security requirements modeling using the Goal-Oriented Requirements Language (GRL) (ITU, 2002) and UMLsec, a security extension to the Unified Modeling Language (Jurjens, 2005; Elshahry, 2005), and its modifications. The fourth section provides some examples of using GRL and UMLsec models for requirements specifications. We conclude in the final section and provide items for further investigation.

SECURITY GOALS AND REQUIREMENTS

The main system security goals to achieve are confidentiality, integrity, availability, and accountability. Confidentiality ensures that only authorized users or applications are allowed to interact with the system. Integrity ensures that critical data has not been changed in an improper way in the system. Availability ensures that the information and/or services are readily available to an authorized user on demand. Accountability ensures that once authorized users access the system, they are accountable for all of their actions (Whitman & Mattord, 2005). Normally, security requirements should not be specified in terms of the types of security mechanisms or controls that are currently used for implementation. To achieve the security goals, security requirements should be identified. These requirements can be structured around the 12 security requirement types identified in Firesmith (2003).

Table 1 maps and shows the contributions of the security requirements to the security goals. For example, survivability,

Table 1. Mapping security requirements to security goals

Security Requirement	Confidentiality	Integrity	Availability	Accountability
Identification Requirements	•	O		O
Authentication Requirements	•	O		O
Authorization Requirements	•	O		O
Immunity Requirements	O	•	O	
Integrity Requirements		•		
Intrusion Detection Requirements	•	O	O	
Intrusion Prevention Requirements	•	O	O	
Non-repudiation Requirements		O		•
Privacy/Secrecy Requirements	•			
Security Auditing Requirements		O		•
Survivability Requirements			•	
Physical Protection Requirements	•	•	•	O
System Maintenance Requirements	O	O	•	
Conformance Requirements	O	O	O	•
• main contribution	O partial contribution			

physical protection, and system maintenance requirements contribute to the system availability security goal.

MODELING SECURITY REQUIREMENTS

Modeling is an important process that can be used for specifying and analyzing requirements. There are many advantages of developing a model before starting the design process. Desirable modeling formalisms are executable and therefore allows the modeler to verify the model and its dynamic behavior before accepting it. An acceptable model is the basis upon which the design process can build. Moreover, correctness of the implemented system can be checked by verifying the conformity of the system to its specified requirements model. There are many existing modeling languages and formalisms. The most famous is the Unified Modeling Language (UML) (OMG, 2003). UML is a de facto standard in the software industry, and it is being generalized to model systems in general (OMG, 2005). However, we are aware of two modeling formalisms that are capable of expressing security requirements in the model. First, the Goal-Oriented Requirement Language (GRL) (ITU, 2002) is a language for supporting goal-oriented modeling and reasoning of requirements, especially for dealing with non-functional requirements such as security and performance. GRL provides the elements to express several concepts appearing during the requirement elicitation phase. Second, UMLsec, an extension to the UML, has been developed by Jurjens (2005) to express security requirements. It is worth mentioning here that another extension to UML, SecureUML, was introduced in Lodderstedt, Basin, and Doser (2002). However, this extension is only to specify role-based access control to support authorization requirements.

Goal-Oriented Requirement Language (GRL)

GRL is a graphical modeling language for capturing NFRs in general. There are three main categories of elements in GRL: intentional elements, links, and actors. The intentional elements in GRL are goals, tasks, softgoals, resources, and beliefs. These elements are used for models that allow answering questions such as why particular behaviors, informational and structural aspects were chosen to be included in the system requirement; what alternatives were considered; what criteria were used to deliberate among alternative options; and what the reasons were for choosing one alternative over the other. GRL supports the reasoning about scenarios by establishing mappings between intentional GRL elements and non-intentional elements in scenario models of the User Requirements Notation—Functional Requirements (URN-FR). Modeling goals and scenarios are complementary and may help identifying further goals, scenarios, and scenario steps important to stakeholders, thus contributing to the completeness and accuracy of the elicited requirements.

A GRL model consists of several goal model structures. Each structure represents a requirement category. For example, Figure 1 shows an e-banking security system as the root of the security requirements. A requirement can