

STS-Tool: Socio-Technical Security Requirements through Social Commitments

Elda Paja, Fabiano Dalpiaz, Mauro Poggianella, Pierluigi Roberti and Paolo Giorgini
Department of Information Engineering and Computer Science - University of Trento
Via Sommarive, 14 - Trento, Italy - 38123
{paja,dalpiaz,poggianella,roberti,giorgini}@disi.unitn.it

Abstract—Security Requirements Engineering (SRE) deals with the elicitation and analysis of security needs to specify security requirements for the system-to-be. In previous work, we have presented STS-ml, a security requirements modelling language for Socio-Technical Systems (STSs) that elicits security needs, using a goal-oriented approach, and derives the security requirements specification based on these needs. Particularly, STS-ml relates security to the interaction among actors in the STS. In this paper, we present STS-Tool, the modelling and analysis support tool for STS-ml. STS-Tool allows designers to model a STS at a high-level of abstraction, while expressing security needs over the interactions between the actors in the STS, and derive security requirements in terms of social commitments—promises with contractual validity—once the modelling is done.

Keywords-Socio-Technical Systems; Security Requirements; Social Commitments

I. INTRODUCTION

In [1], we argued that security issues in Socio-Technical Systems (STSs) are mainly social, arising from the interaction between actors (humans, organisations, and software/hardware), especially when this involves information exchange. The importance of considering security from a social and organisational perspective has already been acknowledged in the literature [2], [3]. Yet, existing approaches offer high-level concepts that are hard to map to requirements (e.g. see [2]) or build upon technical mechanisms (e.g. see [3]). In our view, SRE should start from high-level abstractions, refining them to derive security requirements.

Based on this, we have proposed STS-ml [1] (Socio-Technical Security modelling language), an actor- and goal-oriented security requirements modelling language for STSs. Notably, STS-ml allows actors to express *security needs* over interactions, to constrain the way interaction is to take place. For instance, a patient might require the hospital not to disclose his medical records to other parties.

STS-ml uses the concept of *social commitments* [4] between actors to represent security requirements. In STS-ml, commitments refer to promises with contractual validity actors exchange with one another to guarantee security properties. For instance, the hospital commits to the patient that his medical records will not be disclosed. The important aspect of social commitments is that they have *contractual validity*. Therefore, in STS-ml they are used as a guarantee for the satisfaction of *security needs*.

In this paper, we illustrate STS-Tool ¹, a graphical tool that supports modelling and analysing STSs in terms of the participating actors and their interactions, as well as the automatic derivation of security requirements in terms of *social commitments* once the modelling is performed.

II. STS-ML

STS-ml belongs to the family of goal-oriented requirements engineering frameworks such as Tropos [5] and SI* [3]. It revises the high-level organisational concepts from Tropos, maintaining a minimal set of concepts, including actor, goal, delegation, etc., and uses the concept of social commitments to capture security requirements.

In STS-ml, a commitment refers to a relationship modelling a (contractual) promise made by an actor (*responsible*) to another actor (*requestor*) for the satisfaction of a *security need*. The list of *social commitments* is derived for each *security need* expressed by the stakeholders, and represents the security requirements specification for the system-at-hand. This specification prescribes the security properties stakeholders have to comply with for their interactions (and the STS) to be secure.

A distinguishing feature of STS-ml is that it offers multi-view modelling, that is, interactions among actors are represented by focusing on different perspectives (views) at a time. STS-ml consists of three different views: *social*, *information*, and *authorisation* (Figure 1). The social view represents actors intentionality (have goals to achieve) and sociality (interact with others to achieve goals and exchange information). The information view gives a structured representation of actors' information and documents. Finally, the authorisation view shows the authorisations actors grant to others over information they own or have authority to do so.

The *security needs* are expressed in all the three views, which together form the *operational view*. The *operational view* is automatically mapped to the specification of *security requirements*, which supports the *security needs* expressed in the *operational view*. Currently, STS-ml supports a number of security needs derived by good security practices with a special focus on interaction, such as *non_repudiation*, *no_delegation*, *need_to_know*, *non_modification*, *non_disclosure*, etc.

¹STS-Tool is available for download at <http://www.sts-tool.eu/Downloads.php>

