Chapter 5.3
A Social Ontology for Integrating Security and Software Engineering

E. Yu
University of Toronto, Canada

L. Liu
Tsinghua University, China

J. Mylopoulos
University of Toronto, Canada

ABSTRACT

As software becomes more and more entrenched in everyday life in today's society, security looms large as an unsolved problem. Despite advances in security mechanisms and technologies, most software systems in the world remain precarious and vulnerable. There is now widespread recognition that security cannot be achieved by technology alone. All software systems are ultimately embedded in some human social environment. The effectiveness of the system depends very much on the forces in that environment. Yet there are few systematic techniques for treating the social context of security together with technical system design in an integral way. In this chapter, we argue that a social ontology at the core of a requirements engineering process can be the basis for integrating security into a requirements driven software engineering process. We describe the ***i**** *agent-oriented modelling framework and show how it can be used to model and reason about security concerns and responses. A smart card example is used to illustrate. Future directions for a social paradigm for security and software engineering are discussed.*

INTRODUCTION

It is now widely acknowledged that security cannot be achieved by technological means alone. As more and more of our everyday activities rely on software, we are increasingly vulnerable to lapses in security and deliberate attacks. Despite ongoing advances in security mechanisms and technologies, new attack schemes and exploits continue to emerge and proliferate.

Security is ultimately about relationships among social actors — stakeholders, system users, potential attackers — and the software that are instruments of their actions. Nevertheless, there are few systematic methods and techniques for analyzing and designing social relationships as technical systems alternatives are explored.

Currently, most of the research on secure software engineering methods focuses on the technology level. Yet, to be effective, software security must be treated as originating from high-level business goals that are taken seriously by stakeholders and decision makers making strategic choices about the direction of an organisation. Security interacts with other high-level business goals such as quality of service, costs, time-to-market, evolvability and responsiveness, reputation and competitiveness, and the viability of business models. What is needed is a systematic linkage between the analysis of technical systems design alternatives and an understanding of their implications at the organisational, social level. From an analysis of the goals and relationships among stakeholders, one seeks technical systems solutions that meet stakeholder goals.

In this chapter, we describe the ***i**** agent-oriented modelling framework and how it can be used to treat security as an integral part of software system requirements engineering. The world is viewed as a network of social actors depending on each other for goals to be achieved, tasks to be performed, and resources to be furnished. Each actor reasons strategically about alternate means for achieving goals, often through relationships with other actors. Security is treated as a high-level goal held by (some) stakeholders that need to be addressed from the earliest stages of system conception. Actors make tradeoffs among competing goals such as functionality, cost, time-to-market, quality of service, as well as security.

The framework offers a set of security requirements analysis facilities to help users, administrators, and designers better understand the various threats and vulnerabilities they face, the countermeasures they can take, and how these can be combined to achieve the desired security results within the broader picture of system design and the business environment. The security analysis process is integrated into the main requirements process, so that security is taken into account from the earliest moment. The technology of smart cards and the environment surrounding its usage provides a good example to illustrate the social ontology of ***i****.

In the next section, we review the current challenges in achieving security in software systems, motivating the need for a social ontology. Given that a social modelling and analysis approach is needed, what characteristics should it have? We consider this in the following section. The two subsequent sections describe the ontology of the ***i**** strategic actors modelling framework and outline a process for analyzing the security issues surrounding a smart card application. The last section reviews several areas of related work and discusses how a social ontology framework can be complementary to these approaches.

BACKGROUND

Despite ongoing advances in security technologies and software quality, new vulnerabilities continue to emerge. It is clear that there can be no perfect security. Security inevitability involves tradeoffs (Schneier, 2003). In practice, therefore, all one can hope for is "good enough" security (Sandhu, 2003).