Engineering Secure Systems: Models, Patterns and Empirical Validation

Brahim Hamid, Donatus Weber

PII: S0167-4048(18)30304-3
DOI: 10.1016/j.cose.2018.03.016
Reference: COSE 1324

To appear in: *Computers & Security*

Received date: 14 February 2017
Revised date: 15 February 2018
Accepted date: 31 March 2018

Please cite this article as: Brahim Hamid, Donatus Weber, Engineering Secure Systems: Models, Patterns and Empirical Validation, *Computers & Security* (2018), doi: 10.1016/j.cose.2018.03.016

Engineering Secure Systems: Models, Patterns and Empirical Validation

Brahim Hamid and Donatus Weber[a,b]

[a]IRIT, University of Toulouse,118 Route de Narbonne,31062 Toulouse Cedex 9, France
Email:hamid@irit.fr
Tel: +33 (0)5 6150 2386
Fax: +33 (0)5 6150 4173
[b]Chair for Embedded Systems, University of Siegen, Hoelderlinstrasse 3, D-57076 Siegen, Germany

Abstract

Several development approaches have been proposed to handle the growing complexity of software system design. The most popular methods use models as the main artifacts to construct and maintain. The desired role of such models is to facilitate, systematize and standardize the construction of software-based systems. In our work, we propose a model-driven engineering (MDE) methodological approach associated with a pattern-based approach to support the development of secure software systems. We address the idea of using patterns to describe solutions for security as recurring security problems in specific design contexts and present a well-proven generic scheme for their solutions. The proposed approach is based on metamodeling and model transformation techniques to define patterns at different levels of abstraction and generate different representations according to the target domain concerns, respectively. Moreover, we describe an operational architecture for development tools to support the approach. Finally, an empirical evaluation of the proposed approach is presented through a practical application to a use case in the metrology domain with strong security requirements, which is followed by a description of a survey performed among domain experts to better understand their perceptions regarding our approach.

Keywords: Security, System engineering, Pattern, Meta-modeling, Model driven engineering.

1. Introduction

System and software security engineering (Anderson, 2008; Devanbu et al., 2000; Barnabe et al., 2011) has become a crucial business aspect because organizations are completely dependent on computer-based systems and invest substantial resources in maintaining them. Standards are available for securing IT systems, such as NIST 800-60, and Control Systems (ICS), such as NIST 800-82. Although they give little guidance to software engineers on how to implement them, they should be applied from the early stages of the conception of a system. Most work must be done manually because only a few tools are available to aid in the implementation. This causes extensive work and incurs substantial extra costs. Thus, there is a need to support the engineering of secure system processes with as much automation as possible. Therefore, developers of these systems need to "design for security". This includes defining the current structure of the system, i.e., the system architecture, finding abstract risks and concrete vulnerabilities, and implementing the appropriate countermeasures to mitigate risks and vulnerabilities to meet the security requirements of these systems. Our contribution to this challenge is to make the system security engineering process manageable and understandable through novel methods and tools that ensure that system security solutions are built by design.

Security experts, practitioners and researchers from different international organizations, associations and academia have agreed that for security, "it's not just the code" (Neumann, 2004; M. Howard, 2007; Fernandez, 2013). The most popular and well-known software security vulnerabilities are design issues, particularly architecture design issues. From the system developer perspective, security issues need to be identified early in the first development steps and at the highest levels, primarily in the architecture design stage, where their semantics are clear. When security requirements are determined, architecture and design activities are conducted using modeling techniques and tools for higher quality and seamless development. The integration of security features using this approach requires high expertise for developing the architecture and design and the availability of both application-domain-specific knowledge and security expertise. Because few experts with this diverse set of experiences exist, capturing and providing this expertise by means of security patterns (Yoder and Barcalow, 1998; Schumacher, 2003; Anwar et al., 2006; Hafiz et al., 2007) can enhance system development by integrating them into different stages of systems engineering. Therefore, security solutions are described as security patterns, and the application-domain specific use of these patterns is provided in terms of security technological products that are already established in that domain.

In this paper, we present a model-based approach for engineering secure systems that uses *patterns* to represent security solutions and knowledge, which fosters reuse. This work is conducted within the context of a model-based security and dependability research project, and our collaboration with security-critical system suppliers suggested a need for this work. The security solutions used by security-critical system