Chapter 6
Designing Secure Software by Testing Application of Security Patterns

Takanori Kobashi
Waseda University, Japan

Haruhiko Kaiya
Kanagawa University, Japan

Hironori Washizaki
Waseda University, Japan & National Institute of Informatics, Japan & SYSTEM INFORMATION Co Ltd, Japan & eXmotion, Japan

Takao Okubo
Institute of Information Security, Japan

Yoshiaki Fukazawa
Waseda University, Japan

Nobukazu Yoshioka
National Institute of Informatics, Japan

ABSTRACT

Simply confirming potential threats and vulnerabilities in an early stage of the development process (e.g., the requirement or design phase) is insufficient because software developers are not necessarily security experts. Additionally, even if the software design considers security at an early stage, whether the software actually satisfies the security requirements must be confirmed. To realize secure design, the authors propose an application to design software systems with verification of security patterns using model testing. The method provides extended security patterns, which include requirement- and design-level patterns as well as a new designing and model testing process that uses these patterns. Once developers specify threats and vulnerabilities in the target system in an early stage of development, the method can verify whether the security patterns are properly applied and assess if the vulnerabilities are resolved.

DOI: 10.4018/978-1-5225-6313-6.ch006

INTRODUCTION

Security has become a critical issue as more businesses operate on open networks and distributed platforms. Software must be supported with security measures (Maruyama, Washizaki, & Yoshioka, 2008). Because threats and vulnerabilities within a system cannot be sufficiently identified during the early development stage, security measures must be addressed in every phase of software development from requirements engineering to design, implementation, testing, and deployment. However, creating software with adequate security measures is extremely difficult due not only to the vast number of security concerns, but also the fact that not all software engineers are security specialists.

Patterns, which are reusable packages that incorporate expert knowledge, represent frequently recurring structures, behaviors, activities, processes, or "things" during the software development process. Many security patterns and abstract security patterns have been proposed to resolve security issues (Buschmann, Fernandez-Buglioni, Schumacher, Sommerlad, & Hybertson, 2006) (Lai, Nagappan, & Steel, 2005) (Fernandez, et al., 2018) (Fernandez, et al., 2016) (Fernandez, Washizaki, & Yoshioka, 2016) (Fernandez, Yoshioka, & Washizaki, 2015a) (Fernandez, Yoshioka, & Washizaki, 2015b) (Fernandez, Yoshioka, & Washizaki, 2014) (Fernandez, et al., 2014)(Fernandez, Yoshioka, & Washizaki, 2008). For example, Buschmann et al. (2006) developed 25 design-level security patterns. By referencing these patterns, developers can efficiently realize software with a high security level.

Security patterns, which are a level of abstraction, encapsulate security-related problems and solutions that recur in certain contexts for secure software system development and operations (Maruyama, Washizaki, & Yoshioka, 2008) (Washizaki, 2017) (Fernandez, et al., 2010) (Nhlabatsi, et al., 2010). Since the late 1990's, almost 500 security patterns have been proposed.

Although UML-based models are widely used for design, especially for model-driven software development, whether patterns are applied correctly is often not verified (Maruyama, Washizaki, & Yoshioka, 2008). The lack of systematic guidelines with respect to applications may result in inappropriately applied security patterns. In particular, developers can instantiate security patterns at the wrong places with the incorrect structure. Additionally, properly applying a security pattern does not guarantee that threats and vulnerabilities are resolved. These issues may result in security damage.

To address the aforementioned problems, we propose a method to design and verify security patterns using model testing. Our method extends existing security patterns, formalizes the security and pattern requirements, confirms that the patterns are properly applied, and assesses whether vulnerabilities are actually resolved.