Validating Security Design Pattern Applications by Testing Design Models

Takanori Kobashi, Computer Science and Engineering Department, Waseda University, Tokyo, Japan
Nobukazu Yoshioka, GRACE Center, National Institute of Informatics, Tokyo, Japan
Haruhiko Kaiya, Department of Information Sciences, Kanagawa University, Kanagawa-ken, Japan
Hironori Washizaki, Computer Science and Engineering Department, Waseda University, Tokyo, Japan
Takano Okubo, Institute of Information Security, Yokohama, Japan
Yoshiaki Fukazawa, Computer Science and Engineering Department, Waseda University, Tokyo, Japan

ABSTRACT

Software developers are not necessarily security experts, confirming potential threats and vulnerabilities at an early stage of the development process (e.g., in the requirement- and design-phase) is insufficient. Additionally, even if designed software considers security at an early stage, whether the software really satisfies the security requirements must be confirmed. To realize secure design, this work proposes an application to validate security patterns using model testing. Its method provides extended security patterns, which include requirement- and design-level patterns as well as a new model testing process using these patterns. After a developer specifies threats and vulnerabilities in the target system during an early stage of development, this method can validate whether the security patterns are properly applied and assess if these vulnerabilities are resolved.

Keywords: Component, Model Testing, Security Patterns, Test-Driven Development, UML

1. INTRODUCTION

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

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 have been proposed to resolve security issues (Bschmann, Fernandez-Buglioni, Schumacher, Sommerlad, & Hybertson, 2002). For example, Bschmann et al (2006) propose 25 design-level security patterns. By referring to these patterns, developer can realize software with high security level efficiently.

Although UML-based models are widely used for design, especially for model-driven software development, whether the patterns are applied correctly is often not verified (Maruyama, Washizaki, & Yoshioka, 2008). Therefore, it is possible to apply a security pattern inappropriately. Additionally, properly applying a security pattern does not guarantee that threats and vulnerabilities will be resolved. These issues may result in security damage. Thus, we propose an application to validate security patterns using model testing. Our method confirms that security patterns are properly applied and assesses whether vulnerabilities are resolved. Our research aims to answer the following two Research Questions (RQs):

• **RQ1:** Can our method validate an appropriate application of the security design pattern in a design model?

• **RQ2:** Can our method validate the presence of vulnerabilities identified at the requirement stage before and after applying patterns?

Herein we answer these two research questions. Because the Security Pattern alone does not provide systematic guidelines with respect to applications, we formally extended existing security patterns. Then we proposed a new testing process to validate applied patterns and a tool to support model testing. Our method provides three major contributions:

- New extended security patterns using Object Constraint Language (OCL) expressions, which include requirement- and design-level patterns;
- A new model-testing process based on Test-Driven Development (TDD) to validate appropriate pattern applications and the existence of vulnerabilities using these extended patterns;
- A tool to support pattern application by creating a script to execute model testing automatically.

This paper is organized as follows. Section II describes the background and problems with security software development. Section III describes related works. Section IV details our new method, which integrates the security patterns. Section V applies our pattern to a case study. Section VI describes the threats to validity of our method. Finally, Section VII summarizes this paper.

### 2. BACKGROUND AND PROBLEMS

In this section, we overview common existing techniques for secure design.

#### 2.1. Security Requirement Patterns (SRPs)

The security requirement pattern (SRP) is an existing technique to identify assets, threats, and countermeasures (Kaiya, Okubo & Yoshioka, 2012). A security pattern is reusable as a security package and includes security knowledge, allowing software developers to design secure systems like a security expert. Various types of security patterns exist. For example, SRPs are used at the requirement level, while security design patterns (SDPs), which are described in Section C, are applied at the design stage level.
The Design Principles and Practices of Interoperable Smart Spaces
www.igi-global.com/chapter/design-principles-practices-interoperable-smart/55434?camid=4v1a

Platform Independent Analysis of Probabilities on Multithreaded Programs
www.igi-global.com/article/platform-independent-analysis-of-probabilities-on-multithreaded-programs/103281?camid=4v1a

Maturing Requirements Engineering Process Maturity Models
www.igi-global.com/chapter/maturing-requirements-engineering-process-maturity/28404?camid=4v1a