Chapter 5
Modeling and Re-Evaluating Security in an Incremental Development of RBAC-Based Systems Using B Method

Nasser Al-Mur Al-Hadhrami
Ministry of Education, Oman

ABSTRACT

Incremental software development through the addition of new features and access rules potentially creates security flaws due to inconsistent access control models. Discovering such flaws in software architectures is commonly performed with formal techniques that allow the verification of the correctness of a system and its compliance with applicable policies. In this chapter, the authors propose the use of the B method to formally, and incrementally, design and evaluate the security of systems running under role-based access control (RBAC) policies. They use an electronic marking system (EMS) as a case study to demonstrate the iterative development of RBAC models and the role of the B language in exploring and re-evaluating the security of the system as well as addressing inconsistencies caused by incremental software development. Two formal approaches of model checking and proof obligations are used to verify the correctness of the RBAC specification.

DOI: 10.4018/978-1-5225-6313-6.ch005

Copyright © 2019, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

INTRODUCTION

Critical systems, including e-commerce, enforce security policies that preserve the availability, integrity, and secrecy of data. Defects in applied security policies for a given system, e.g., ambiguous properties or inconsistent access control models, result in unreliable security of the system. Furthermore, from the process of incremental software development, the addition of new features and insertion of new access rules may render the access control models inconsistent and create security flaws. Therefore, exploring and addressing such flaws in software architectures is required to preserve the reliability of a system. A common approach to study and analyse security is to formally specify the system and its properties in models that allow for the verification of their correctness and compliance to applicable policies, such as Role-based Access Control (RBAC) policies (Ferraiolo & Kuhn, 1995).

RBAC is one of the most effective security models adopted in recent years that facilitates the administration of security in large organisations. The principal motivation behind RBAC models is that users are not directly granted access to an enterprise's objects. Instead, access permissions are administratively associated with roles, and users are administratively assigned to appropriate roles. This mechanism simplifies the management of authorisations and provides flexibility in specifying and enforcing security policies, particularly in dynamic systems (Ferraiolo & Kuhn, 1995). Evaluation of the correctness and consistency of such policies is essential to ensure the reliability and robustness of dynamic systems.

We discuss the application of the B language (Abrial, 1996) to incrementally develop and re-evaluate the security of an Electronic Marking System (EMS) running under RBAC policies. With B specifications, we study the impact of iterative development on the system's specification of access rights for subjects to resources modelled as an RBAC. The access rights include, for example, that teachers can access the system to add, edit or delete marks, and students have the authorisation to submit reports and view their grades. Such a system requires reliable assurances that the RBAC model is consistent and complies with a set of security policies.

The B language is a formal method based on *set theory* and *first-order logic* used to model and refine a system's specification (Schneider, 2001) using a special notation, i.e., language, called Abstract Machine Notation (AMN). The model development process creates multiple proof obligations that guarantee the correctness of the model and the desired properties (invariants) that the model must preserve. Proving the obligations, verifying the properties, and simulating the model are functions commonly supported by tools such as ProB (Leuschel & Butler, 2003).

This chapter is organised as follows. In Section 2, we provide overviews of the RBAC model, its properties, and refer to related work on applying formal specifications to implement RBAC policies. In Section 3, we overview the B method

30 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/modeling-and-re-evaluating-security-in-an-incremental-development-of-rbac-based-systems-using-b-method/221714

Related Content

Prediction of Customer Review's Helpfulness Based on Feature Engineering Driven Deep Learning Model

Surya Prakash Sharma, Laxman Singhand Rajdev Tiwari (2023). *International Journal of Software Innovation (pp. 1-16).*

www.irma-international.org/article/prediction-of-customer-reviews-helpfulness-based-on-feature-engineering-driven-deep-learning-model/315734

Workload Classification: For Better Resource Management in Fog-Cloud Environments

Zahid Razaand Nupur Jangu (2022). *International Journal of Systems and Service-Oriented Engineering (pp. 1-14).*

www.irma-international.org/article/workload-classification/297135

Building Secure Software Using XP

Walid Al-Ahmad (2011). *International Journal of Secure Software Engineering (pp. 63-76).*

www.irma-international.org/article/building-secure-software-using/58508

Strategies for Static Tables

Dean Kelley (2009). *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications (pp. 148-160).*

www.irma-international.org/chapter/strategies-static-tables/21068

Software Defects Prediction Model with Self Improved Optimization

Shantappa G. Gollagi, Jeneetha Jebanazer Jand Sridevi Sakhamuri (2022). *International Journal of Software Innovation (pp. 1-21).*

www.irma-international.org/article/software-defects-prediction-model-with-self-improved-optimization/309735