An Incremental B-Model for RBAC-Controlled Electronic Marking System

Nasser Al-hadhrami, Ministry of Education, Nizwa, Oman
Benjamin Aziz, University of Portsmouth, Portsmouth, UK
Lotfi ben Othmane, Fraunhofer Institute for Secure Information Technology, Darmstadt, Germany

ABSTRACT

The incremental development of software through the addition of new features and the insertion of new access rules potentially renders the access control models inconsistent and creates security flaws. This paper proposes modeling Role Based Access Control (RBAC) models of these software using the B language and re-evaluating the consistency of the models following model changes. It shows the mechanism of formalizing RBAC policies of an Electronic Marking System (EMS) using B specifications and illustrates the verification of the consistency of the RBAC specification, using model checking and proof obligations. In addition, it shows how to address inconsistencies that result from incremental specification of system’ architectures.

KEYWORDS

Incremental Software Development, Model Checking, Proof Obligations, Role-Based Access Control Constraints

1. INTRODUCTION

Critical systems, including e-commerce systems, enforce security policies that preserve availability, integrity, and secrecy of data. Defects in applying security policies of a given system, e.g., ambiguous properties and inconsistent access control models, makes the security of the system unreliable. A common approach to address such inconsistencies is to formally specify the given system and its properties in models which allows to verify the correctness of the models and their compliance to applicable policies. This applies to formalizing accesses to systems as Role-Based Access Control (RBAC) policies.

We discuss in this paper the iterative development of Electronic Marking System (EMS) and its impact on access rights of subjects to resources modeled as an RBAC model. The access rights include, for example, that teachers can access the system for the purpose of adding, editing or deleting marks, whereas students have the authorization to submit reports and view their grades. Such systems require reliable assurance that the RBAC model is consistent and it complies with a set of security policy. To address this, we use the language B (Abrial, 1996) to model the system and verify its properties. The model development process creates a number of proof obligations that guarantee the correctness of the model and the desired properties (invariant) that the model shall preserve. Proving

DOI: 10.4018/IJSSE.2016040103

Copyright © 2016, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
the obligations, verifying the properties, and simulating the model are functions commonly supported by tools such as ProB (Leuschel & Butler, 2003).

This paper is an extension to our earlier workshop paper (Al-hadhrami et al. 2015). It is organized as follows. In Section 2, we provide overviews of the RBAC model and its properties. In Section 3, we refer to some related works on using formal specifications to implement RBAC policies, and in section 4, we provide overview of the B method used to model formally our EMS system specification running under RBAC policies. In Section 5, we introduce our EMS system without any policies. In Section 6, we describe the basic RBAC model, namely RBAC0 of the system’s access policies and we discuss the weaknesses and inconsistencies of the RBAC0-controlled model of EMS. In Section 7, we further develop the model to be able to run under RBAC2 policy constraints. We focus mainly on types of such constraints; separation of duties and cardinality constraints. In Section 8, we model the EMS under role hierarchies (i.e. RBAC1). Finally, in Section 9, we conclude the paper discussing prospects for future work. We have also two appendices for further reading: Appendix A shows how to model operations for User Assignment (UA) and revoking assignments under the policies defined in the three versions of the RBAC model. In Appendix B, we discuss two approaches for formal verification (i.e. model checking and theorem proving) to ensure that the system is consistent and secure.

2. ROLE BASED ACCESS CONTROL

RBAC is an efficient and safe role-based access control model (Ahn & Hu, 2007). Began in 1970s with multi-user and multi-application, and has rapidly evolved in the last three decades as a technology for applying a high level security in large-scale systems. The pivotal idea behind RBAC model is that permissions are associated with roles, and users are administratively assigned to proper roles. This mechanism ensures that only authorized users can perform some functions on some data/resources (Ferraiolo & Kuhn, 2009). Figure 1 shows that users are not directly mapped into permissions of accessing some resources, but to specific roles which have to be previously assigned to those permissions.

Unlike traditional lattice-based access control policies, such as Discretionary Access Control (DAC) and Mandatory Access Control (MAC), security administration for RBAC models can be simply implemented through using roles to organize access privileges, because there are many users who can be assigned to a particular role, and the permissions for that role should be previously defined for once. For example, it is easy to move a user to a new functional role within an organization through just removing his association with the old role and assigning him to a new one without a need to manipulate/change the permissions of the old role. Another benefit of RBAC models is that they implement the principle of ‘least privileges’ which states that users should have only the privileges they need to do their tasks and no more (Yu & Brewster, 2012).

Figure 1. The concept of RBAC security policy
Parallel Online Exact Summation of Floating-point Numbers by Applying MapReduce of Java8
www.igi-global.com/article/parallel-online-exact-summation-of-floating-point-numbers-by-applying-mapreduce-of-java8/176665?camid=4v1a

Formal Specification and Implementation of Priority Queue using Stream Functions
www.igi-global.com/article/formal-specification-and-implementation-of-priority-queue-using-stream-functions/103282?camid=4v1a