Chapter 49
RBAC with Generic Rights, Delegation, Revocation, and Constraints

Jacques Wainer
University of Campinas, Brazil

Fabio Negrello
University of Campinas, Brazil

Igor Ribeiro de Assis
University of Campinas, Brazil

ABSTRACT

This chapter presents R+DRC, an extension of the Role-based Access Control (RBAC) model. R+DRC allow for defining constraints, for example to enforce different forms of separation of duties, and the right of overriding a constraint. The model also defines delegations, and two forms of revocations. The model is discussed within the framework of modeling the access control of an hospital. Algorithms are provided for the more complex actions.

INTRODUCTION

Consider the following description of some of the rights in a hospital. The doctor who admits a patient can prescribe drugs, ask for exams, schedule procedures, and discharge the patient in the non emergency ward of the hospital. A doctor may have a team to whom he delegates some or all the rights regarding the patients he admitted. The doctor may also delegate different rights to different members of the team, so some team member may have the right to prescribe medication to the patients but not to discharge them. The doctor may also assemble an ad hoc team of other physicians to deal with a particular patient, and delegate different rights regarding that patient to different physicians in the ad hoc team.

If the patient dismisses his doctor, or if no one who has the right to discharge the patient can be reached after some reasonable effort, then the chief medical officer of the hospital may discharge the patient. The chief medical officer may also del-

DOI: 10.4018/978-1-4666-2919-6.ch049

egate this right to exceptionally discharge a patient to a few other trusted colleagues, if he knows he will be unavailable for some time.

This scenario presents some challenges to a standard ANSI-RBAC modeling (ans, 2004). Most of the rights discussed above are not rights that are attached to roles such as the right to admit a patient which is attached to the role of physician. The right to prescribe drugs to patient x is not attached to the role of a physician but to the specific members of the medical team responsible for the patient x. Furthermore, some of the rights discussed above are not attached to a particular patient, but are in fact generic - the chief medical officer has the right to discharge *any* patient (and has the right to delegate this right). In ANSI-RBAC, rights are necessarily attached to objects in what is called *permissions*. Thus, in standard RBAC, for each new patient admitted, a new permission of discharging him or her should be created and associated to the chief medical officer role.

The other scenario involves a clinical reviewing board. If there are any questions regarding a patient's treatment, a reviewing board of senior physicians may be called for. The reviewing board analyzes the actions taken on behalf of a patient, and thus no one that acted in the patient's treatment can be a member of the reviewing board. This is an example of a *separation of duties* constraint. Furthermore, if reviewing boards are infrequent, the hospital may impose a further restriction that if physician A served on a reviewing board of a patient's case in which physician B was involved, then B cannot serve in the reviewing board of a patient's case in which A was involved. We call this *mutual separation of duties*. But if it becomes hard to select physicians given the mutual separation of duties policy, the chief medical officer can assign physicians to a review board so that it violates the mutual separation of duties, but not the separation of duties rule.

This example illustrates that violating constraints is also a right that some particular roles may have, in order to guarantee that the work should proceed. Violations of constraints is particularly important in business processes, where some constraints represent desirable but not necessarily required rules.

Finally, let us consider the following scenario in a large engineering company. In this company, people are added to projects as it becomes clear during the project development that their particular expertise is needed.

Now let us consider the situation in which engineer A invites B to work in a part of the project because B is one of the specialists in radiation calculations. Or in the terms of this chapter, A delegates to B not only the access to the project, but also the right to delegate it further, that is, B also has the right to invite other engineers to the project. B delegates access to C which is one of the specialists in radiation safety regulations. Some weeks later, D also delegates to C access to the project, because C is also a specialist in fire safety regulations. Now let us suppose that A leaves the company, and thus all delegations made by A must be evaluated by his substitute, and the safest way to proceed is to revoke all delegations made by A and add new delegations as the substitute approve them. But it is desirable that the revocation of the delegations causes the least changes as possible as not to totally halt the project. In particular, the standard, time-stamp based algorithm for delegation would revoke A's delegation to B, and B's delegation to C, but a more careful algorithm could realize that C's access can be justified by D's independent delegation.

The three scenarios above illustrate the issues we will tackle in this chapter. The first hospital scenario raises the issues of:

- **Generic Rights:** Rights that apply to any object of a class.
- **Direct Rights:** Rights that are directly associated to users, and are not mediated through roles. Usually these rights come about from delegations but there are other forms.