Positive and Negative Authorizations to Access Protected Web Resources

Sylvia Encheva[1] and Sharil Tumin[2]

[1] Stord/Haugesund University College, Bjørnsonsg. 45, 5528 Haugesund, Norway
sbe@hsh.no
[2] University of Bergen, IT-Dept., P.O. Box 7800, 5020 Bergen, Norway
edpst@it.uib.no

Abstract. In this paper we present a model that can prevent conflict situations caused by applying both positive and negative authorizations for access to a resource. Such conflict situations may occur if an organization has decentralized administration, and/or several collaborating organizations have access to one resource and some of them apply positive authorizations while others apply negative authorizations. The proposed solution involves Belnap's logic.

Keywords: Collaboration, positive and negative authorization.

1 Introduction

A number of computer-based access control systems apply positive authorizations. Such authorizations define those accesses that are going to be allowed [1]. Authorization models supporting positive authorization apply closed policy, i.e. access is allowed to a few users only.

A serious problem with this approach is that a particular user without a given authorization from this closed policy can obtain an authorization from a different resource manager. Suppose a resource at an educational institution is accessible by lecturers and students. This institution has decentralized administration, i.e. lecturers and students are managed by different resource managers. Then a user, who is a student in one subject and a lecturer in another one, can lack authorization to access a resource as a student but might be able to obtain the desired authorization as a lecturer.

Authorization models supporting negative authorization apply open policy, i.e. accesses are to be allowed to all but a few users. Examples of negative authorizations are an approval of a bank card transaction in which the customer's account number is compared against a list of cancelled accounts numbers and verification systems in which only poor credit risks are noted in the credit check. A credit that is not negatively reported is assumed to be verified. Other applications are discussed in [2], [4], and [9].

Negative authorizations are often used because they give opportunities to include exceptions [7]. Without involvement of exceptions one should considerably increase the number of authorizations, and thus make the management of authorizations more complicated. Following the example with lecturers and students

T. Enokido, L. Barolli, and M. Takizawa (Eds.): NBiS 2007, LNCS 4658, pp. 79–87, 2007.
© Springer-Verlag Berlin Heidelberg 2007

we consider a case where all lecturers are granted an authorization to a resource with the exception of the one who is also a student. Without negative authorizations, one should satisfy the requirement by giving a positive authorization to each lecturer except one. With help of negative authorizations, this situation can be resolved by granting a positive authorization to the group of lecturers and a negative authorization to the one who is a also student.

We present a model that prevents conflicts generated by applying positive and negative authorizations to users accessing resources in a large networked system.

The rest of the paper is organized as follows. Related work, basic terms and concepts are presented in Section 2. The management model is described in Section 3. The system architecture is discussed in Sections 4, 5. The paper ends with a description of the system implementation in Section 6 and a conclusion in Section 7.

2 Background

A formal model of role based access control (RBAC) is presented in [12]. Permissions in RBAC are associated with roles, and users are made members of appropriate roles, thereby acquiring the roles' permissions. The RBAC model defines three kinds of separation of duties - static, dynamic, and operational. Separation of duties was discussed in [7], [13] and [17]. A framework for modeling the delegation of roles from one user to another is proposed in [3]. A multiple-leveled RBAC model is presented in [10]. The design and implementation of an integrated approach to engineering and enforcing context constraints in RBAC environments is described in [18] and [19].

While RBAC provides a formal implementation model, Shibboleth [16] defines standards for implementation, based on OASIS Security Assertion Markup Language (SAML). Shibboleth defines a standard set of instructions between an identity provider (Origin site) and a service provider (Target site) to facilitate browser single sign-on and attribute exchange.

The semantic characterization of a four-valued logic for expressing practical deductive processes is presented in [6]. In most information systems the management of databases is not considered to include neither explicit nor hidden inconsistencies. In real life situation information often come from different contradicting sources. Thus different sources can provide inconsistent data while deductive reasoning may result in hidden inconsistencies. The idea in Belnap's approach is to develop a logic that is not that dependable of inconsistencies. The Belnap's logic has four truth values 'T, F, Both, None'. The meaning of these values can be described as follows: an atomic sentence is stated to be true only (T), an atomic sentence is stated to be false only (F), an atomic sentence is stated to be both true and false, for instance, by different sources, or in different points of time (Both), and an atomic sentences status is unknown. That is, neither true, nor false (None).

A user is defined as a valid domain identity at a particular organization O_i. A group is a set of users. A resource defines a set of protected Web objects.