An Administration Model of DRBAC on the Web*

Qi Li
Graduate School of Chinese Academy of Sciences And ERCIST.COM, China
qi.li@ercist.com

Jingpu Shi
Electrical and Computer Engineering Department Rice University, USA
jingpu@rice.edu

Sihan Qing
Graduate School of Chinese Academy of Sciences And ERCIST.COM, China
qsihan@ercist.iscas.ac.cn

Abstract

Role-based Access Control (RBAC) model has been widely deployed for web security in recent years. In some large-enterprise-wide situations, however, this model is difficult to manage due to huge amount of users, roles and interrelationships. As a result, the applications of this model are greatly limited. To address this problem, we present in this paper a Decentralized RBAC model (DRBAC) by proposing a new concept of groups and by introducing non-trivial modifications to the user-role assignment of the existing models.

1. Introduction

Role-based Access Control (RBAC) model was found to be the most attractive solution for providing information security on the web, which is essential for many web applications such as e-commerce and e-government. The RBAC model has been shown to be "policy-neutral" in the sense that, using hierarchies and constraints, a wide range of security policies can be expressed, including Discretionary Access Control (DAC), Mandatory Access Control (MAC) and User-specific Access Control (USAC) [7]. To fully take advantage of this property in a large enterprise, where the number of roles and users in a RBAC system vary from tens to thousands, it is essential to manage these components. Many researchers provide various administration models for RBAC [2, 5, 6, 7], which, however, share the same weakness when applied to a large enterprise in that it is difficult to manage a large number of RBAC elements. In this paper, we propose a Decentralized RBAC (DRBAC) model to overcome this shortcoming by adding a new concept of groups and redefining the user-role assignment in [2, 5, 6, 7].

This paper is organized as follows. Section 2 introduces the proposed DRBAC model. Section 3 presents our administration strategy. Section 4 outlines the model implementation. We conclude this paper in Section 5.

2. The DRBAC model

2.1. Model differences from RBAC

The DRBAC model we propose introduces the concept of groups, based on which user-role assignment is implicitly or explicitly conducted. The essential difference between groups and roles is that a group is a collection of users with the same security attributes while a role is a collection of permissions. We provide these two levels of the DRBAC model, which respectively represent the system class administration model associated with central control over user-role assignment, and the group class administration model associated with decentralized control over user-role assignment. In the rest of this paper, we primarily focus on the second-level decentralized model.

2.2. Model description

Our model is composed of the basic DRBAC model, Role Hierarchies (RH) and Constraints. These three parts are corresponding to RBAC0, RBAC1 and RBAC2 in [4] respectively, as shown in Fig. 1. In the rest of this section, we introduce the three parts.

The basic DRBAC consists of five components: U (users), G (groups), R (roles), P (permissions) and S (sessions). We also define User-Group mapping

* Supported by the Beijing Natural Science Foundation under Grant No. 4052016; the National Natural Science Foundation of China under Grant No. 60083007; the National Grand Fundamental Research 973 Program of China under Grant No.G1999035802.

(UM) responsible for associating users with groups and *IUA* responsible for the assignment between roles. *IUA* provides the mechanism through which a group member is assigned the roles and gets the permission. It is implemented by mapping the users to groups *(UM)* and assigning role to groups (*GA*). These roles we mentioned above are called Default Group Roles (*DR*). The following definitions (or requirements indeed) formalize the basic DRBAC.

Definition 1: The DRBAC model assignment relations:

- $PA \subseteq P \times R$, a many-to-many permission to role assignment relation
- $UM : U - G$, a single-to-many user to group mapping relation
- $GA \subseteq G \times R$, a many-to-many Group to role assignment relation
- $EUA \subseteq U \times R$, a explicit user to role assignment relation
- $IUA \subseteq U \times GA$, a implicit user to role assignment relation

$$IUA = \left\{ \begin{array}{l} (r:R, ga:GA) \mid \exists (g:G, u:U) \cdot \\ [(r,(u,g) \in EUA)] \wedge [(u,g) = um] \\ \wedge [(g,r) \in ga] \end{array} \right\}$$

- $UA = EUA \cup IUA$,a many-to-many user role assignment relation
- $user : S - U$, a function mapping each session s_i to the single user $user(s_i)$ (constant for the session's lifetime) and
- $permissions : R \rightarrow 2^P$, a function mapping a role to a set of assigned permissions and
- $roles : S - 2^R$, a function mapping each session s_i to a set of roles $roles(s_i) \subseteq \{r \mid (user(s_i), r) \in UA\}$ (which can change with time) and session s_i , has the permission

$$\cup_{r \in roles\ (s_i)} \{ p \mid (p, r) \in PA \}$$

When a user login an application, he creates a session in DRBAC and activates a subset of the user's roles including some of user's roles themselves and some of the default group roles. Users can also change the active roles in a session. And the session can be terminated by the user or by the system due to a long idle duration.

Definition 2: The same definition in [1, 4] is used to describe the DRBAC role hierarchies.

Definition 3: DRBAC requires that a collection of constraints must be used to determine whether values of various components of RBAC are acceptable.

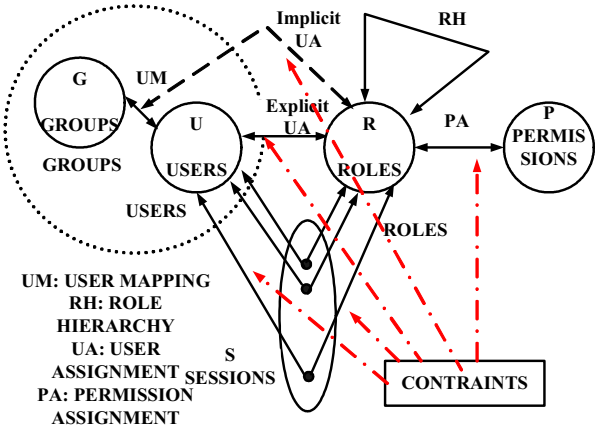

Figure 1 DRBAC Model

3. The DRBAC Administration Model

The goal of the management model, as shown in Fig. 3, is to impose restrictions on points. The first level model regulates who can add or remove a role to which user, and the second level controls which permission can be added or removed from the operation sets of a role. In the first-level decentralized scheme, we directly extend the model in [5]. In the rest of this section, we explain the second-level scheme in detail, which is critical for the complex administration of *UA*.

3.1 DRBAC Grant Model

Definition 4: A prerequisite condition in the second-level model is defined as a Boolean expression using both $\wedge$ and $\vee$ operators on terms of the form x and $\overline{x}$, where x is a regular role (i.e., $x \in R$). It is evaluated for the user u by interpreting x to be true if there exists a administrative user au and the group administrative role a, and there exists

$$(\exists x' \geq x) \quad [(u,x') \notin UA] \wedge [(au,x') \in UA] \wedge$$
$$[(u,g) \in UM] \wedge [(au,g) \in UM] \wedge [(a,au) \in PA]$$

For a given set of roles R, let CR denote all possible prerequisite conditions that can be formed using roles in R.

To illustrate the prerequisite conditions, Table 1 shows the simplest prerequisites. We put a '@' in

front of the group names to distinguish the role from group names.

In Table 1, all users are members of group ACC. Note that @ACC only means that the user is a member of group ACC, although it means differently in [ARBAC02]. ACC has the role assignment relation with role ACCEE. Let A be a member of the ACCER and B be a member of ACCEE. A satisfies the prerequisite condition to assign the B any of the A1, A2 and A3 but not the A4 role.

Table1. An Example of Prerequisite Condition

Role	Rereq. Condition	Role Range
GUEST	@ACC	[A1,A1]
ACCEE	@ACC	[A1,A2]
OTHEE	@OTH	[A1,A2]
ACCER	@ACC ∧ A3 ∧ AM	[A1,A4]

As shown in Table 1, the AM role is the administrative role of group ACC, which is a parallel role and does not have the partial relation with A1, A2 and A4 or A1, A3 and A4. Fig. 2 illustrates the relation between them.

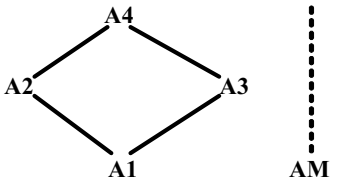

Figure 2 an example role hierarchy

Authorized user-role assignment in the second-level model has the following relations.

Definition 5: The second-level decentralized administrative model controls user-roles assignment according to the relation $can\_assign_2 \subseteq CR \times 2^R$.

The above definitions formalize the relation as $can\_assign(x, y, \{z\})$, which means that a member of the administrative role x (or an administrative role senior to x) can assign a user to be a member of regular z if the user satisfies the prerequisite condition.

Let *A* be a member of ACCER, *B* be a member of ACCEE and C be a member of OTHEE according to Table 1. *A* cannot assign *C* to any of the roles.

3.2 DRBAC Revocation Model

User revocation in DRBAC is controlled by can_revoke. Fig. 3 illustrates the two revocation models in DRBAC architecture.

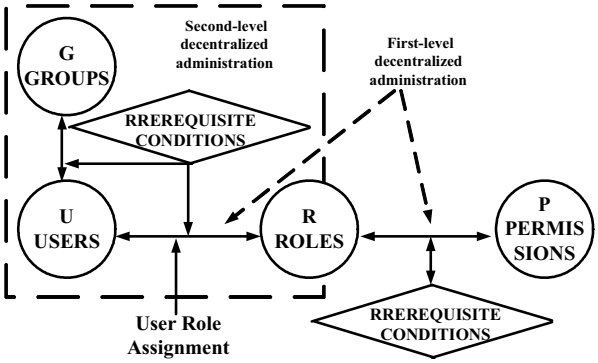

Figure 3 DRBAC administration model

Definition 6: Decentralized administrative model controls user-role revocation by means of the relation $can\_revoke \subseteq R \times 2^R$, where R denotes the administrative roles including system-level and group-level administrative roles [5].

This definition has the similar feature with can_assign relation. For example, $can\_revoke(x, \{z\})$ means that an administrative member of role x (or a administrative role senior to x) can revoke the member of role z.

DRBAC provides a second-level revocation model in which the user with membership in decentralized administrative role handles revocation in the group.

Definition 7: Second-level revocation model. Suppose that a user U is a member of group G and is assigned the role x if $(U, x) \in UA$, $(U, G) \in UM$. Another user AU is also the member of Group G and is assigned the role x' if $x' > x$, $(AU, x') \in UA$, $(AU, G) \in UM$.

- Ineffective / Effective Revocation: Let A be an explicit membership of a group-class administrative user of Group g and have a session with the roles $R = \{r_1, r_2, ..., r_k\}$. Let A try to revoke B from role x. If x is not the element of collection R ($x \notin R$), this operation has no effect. Otherwise if there exists $x \in R$, $(A, g) \in UM$, $(B, g) \in UM$ revokes B's membership in x.

Effective revocation of B 's membership in x requires that A not only has the administrative role over the group with which B has direct mapping relation, but also is the member of role x . Otherwise the operation is an ineffective revocation.

Let us consider the example of revocation in table 2 and interpret it in context of the hierarchies of Fig. 3. And table 4 illustrates the effect of revocation.

Table2. An Example of can_revoke

Role	Role Range
GUEST	[A1,A1]
ACCEE	[A1,A2]
SYSAD	[A1,A4]
ACCER	[A1,A4] $\wedge$ AM

Table3. An Example of revocation effect

User	A1	A2	A3	A4	R	Member of ACC
B	Yes	Yes	No	No	No	Yes
C	Yes	Yes	Yes	No	Yes	Yes
D	Yes	Yes	Yes	Yes	No	Yes
E	Yes	Yes	Yes	No	Yes	No

In the second-level revocation, let A be the manager of group ACC. A cannot revoke C from Role R and cannot revoke E from A1, A2 and A3 because of the administration scope of A.

4. Implementation of the Administration Model on Web

We have implemented and deployed our model on the web including functions for creation and deletion of groups, operations to assign and de-assign permissions and users to a role, and implicit and explicit user role assignment. Our implementation shows that this DRBAC model overcomes many difficulties many other models have when applied in a large enterprise. A set of RBAC models [2, 3, 4, 5, 6] only provide a single user-role association and, therefore, it is tedious for these models to configure every user-role assignment. The models in [8] have many limitations in that the role in the user pool must have partial relation, which proposes great constraints on user-role assignment in a situation where there are many diverse roles. In contrast, our implemented DRBAC introduces the concept of groups, which can be registered into the proper role set. As a result, there is less complicated requirement and the user assignment becomes simpler. Our implementation shows that the DRBAC model significantly improves the administrative management of user-role assignment, especially in a large enterprise web application.

5. Conclusion

In this paper, we present an improved administration model for user-role assignment in a modified RBAC model (DRBAC). The main advantage of the model it provides administration convenience. It does not have to be used in a highly central control environment. And it provides two levels of administration models for user-role assignment and reduces the complexity of administration of RBAC system. Our deployed implementation of this model shows that this DRBAC model greatly simplifies the complexity of RBAC models in large-enterprise applications.

6. References

[1] D. Ferraiolo, and R. Sandhu, and S. Gavrila, and D. Kuhn and R. Chandramouli, "Proposed NIST standard for role-based access control", *ACM Transaction on Information and System Security*, 2001, pp. 224-274.

[2] F. Dridi, B. Muschall, and G. Pernul, "Administration of an RBAC system", *Proceeding of the 37th Hawaii International Conference on System Sciences (HICSS'04)*, Big Island, Hawaii, 2004.

[3] J.S. Park, R. Sandhu, and G.J. Ahn, "Role-Based Access Control on the web", *ACM Transactions on Information and System Security*, 2001, pp. 37-71.

[4] R.S. Sandhu, E.J. Coyne, H.L. Reinstein, and C.E. Youman, "Role-Based Access Control Model", *IEEE Computer*, 1996, pp. 38-47.

[5] R. Sandhu, and J.S. Park, "Decentralized User-Role Assignment for Web-Based Intranets", *Proceeding of 3rd ACM Workshop on Role-Based Access*, Fairfax, VA, 1998, pp. 1-12.

[6] R. Sandhu, V. Bhamidipati, and Q. Munawer, "The ARBAC97 Model for Role-Based Administration of Roles", *ACM Transactions on Information and System Security*, 1999, pp. 105-135.

[7] S. Oh, and R. Sandhu, "A Model for Role Administration Using Organization Structure", *Proceeding of the 7th ACM Symposium on Access Control Models and Technologies (SACMAT'02)*, Monterey, CA, 2002, pp. 155-162.