RESEARCH PAPER | Engineering | Volume : 3 | Issue : 3 | March 2013 | ISSN - 2249-555X

Study of an RBAC System

KEYWORDS

Shuriya. B
RVS Faculty of Engineering, Coimbatore

Dr. S. Sumathi
PSG College of Technology, Coimbatore

ABSTRACT *Role-based access control is a widely used for accessing the role. In the organizations, the RBAC policy is managed by many administrators. An administrative role based access control (ARBAC) policy tells how each administrator may change the RBAC policy. It is often difficult to fully understand the function of an ARBAC policy by simple inspection, because sequences of changes by different administrators may interact and yields unexpected results. ARBAC policy allows roles and permissions to have parameters which are used to maintain the organization. In this paper, we are have studied and analyzed the ARBAC, which will be used for proceeding my work in visual role mining*

1 INTRODUCTION

Each business process includes a number of tasks that need to be executed in order [1,2,3]. In this the access permissions grant the right to do a certain task. Therefore, the users participating in a workflow must have the permissions that are needed to execute the corresponding tasks. Therefore, RBAC directly supports the principle of least privilege because each user can be assigned to the particular roles, and owns the exact number of permissions, that are needed to perform his/her duties. The engineered roles also tend to change significantly slower than the assignment of individuals to these roles. Thus, establishing roles as an abstraction mechanism facilitates the administration of permissions. To enforce the process definitions and access control policies in an information system, the corresponding models need to be mapped to the software platform respectively. However, a number of sophisticated approaches exist that allow for the formal specification and analysis of process related access control policies and constraints [4,5,17]. The corresponding modeling support for software systems is largely missing. In this paper, we study and analysis to model processes and process-related RBAC models. This paper is motivated by to develop the visual role based model of RBAC administration and the requirements that come up in real deployment.

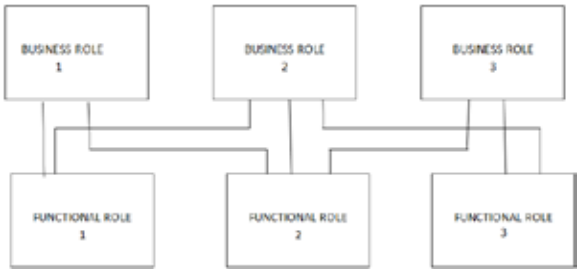

Figure 1: Role Hierarchy

2 ROLES ACCESS

Access is the ability to use, change, or view something with a computer resource. Access control is the means by which the ability is explicitly enabled or restricted in some way. Computer based access control can prescribe not only who or what process may have access to a specific system resource, but also the type of access that has been permitted. These controls may be implemented in computer system or in the external devices. With role-based access control, access decisions are based on the roles that individual users has in part of an organization. Users take on assigned roles. The process of defining roles should be based on the thorough analysis of how an organization operates and also include input from a wide spectrum of users in an organization [7,8,10].Access rights are grouped by role name, and the use of resources is restricted to individuals. The authorization is given to the users to assume the associated role. For example, the role of researcher can be limited to gathering detail study about the particular subject and domain. The use of roles to control access can be an effective means for developing and enforcing enterprise-specific security policies, and also for streamlining the security management process.

3 USERS AND ROLES

Under the RBAC framework, users are granted membership into roles based on their functionalities and responsibilities that have been allocated in the organization. The operations that a user is permitted to perform are based on the user's role in an organization. User membership into roles can be revoked easily and new memberships established as job assignments are made. Role associations can be established when new operations are instituted and old operations can be deleted as organizational functions change. This simplifies the administration and management of privileges; roles can be updated without updating the privileges for every user on an individual basis efficiently. When a user is associated with a role, the user can be given less privilege than is necessary to perform the job. This concept of least privilege requires identifying the user's job functions, determining the minimum set of privileges required to perform that function, and restricting the user to a domain with those privileges. In less precisely controlled systems, this is often difficult or costly to achieve role based access control. Someone assigned to a job category may be allowed more privileges than needed. Since many of the responsibilities overlap between job categories, maximum privilege for each job category. This may cause unlawful access.

4 ROLES AND ROLE HIERARCHIES

RBAC adds the notion of roles as a level of indirection/Direction between users and permissions. Roles are created based on job functions and/or qualifications of the users. Permissions are assigned to roles based on the requirements of job functions and/or qualifications of the user. Users are made members of roles based on their job responsibilities and/or qualifications, thereby accessing permissions assigned to those roles [14,15,18,20]. Roles may be organized into a hierarchy, which defines a partial order among roles that have assigned. We use r1 and r2 to denote that r1 is dominated by r2, and say that r1 is more junior to r2, and r2 is more senior to r1. This means r2 inherits all the permissions that are assigned to r1, and all users who are members of the r2 are also members of r1. Several existing approaches to RBAC administration use role hierarchies to specify the domain of the administration. Therefore, role hierarchies play a key role in the study of RBAC administration. The more specialized roles inherit permissions assigned to the generic roles, and may be assigned an additional permissions. Functional roles are created by the managers, who control resources and also assign permissions for their resources [12,14]. Business roles

Volume : 3 | Issue : 3 | March 2013 | ISSN - 2249-555X

are created by role administrators, who determine what the functional roles are needed for each business role [13,15]. This results in a two-level role hierarchy. Figure 1 shows the role hierarchy.

5 ROLES AND OPERATIONS

Organizations can establish the rules for the association of operations with roles in hierarchy. For example, a researcher can allowed to have study at their particular domain under the university and also can access the university department results and he/she is not allowed to access other scholar's details [6,9,11]. An operation represents a unit of control that can be referenced by an individual role and subject to regulatory constraints within the RBAC framework [16,19]. An operation can be used to capture security relevant details or constraints that cannot be determined by a simple mode of access. The RBAC framework provides administrators with the capability to regulate who can perform what actions, when, from where, in what order, and in some cases under what relational circumstances. Only those operations that need to be performed by members of a role are granted to the role in RBAC. Granting of user membership to roles can also be limited. Some roles can only be occupied by a certain number of employees at any given period of time. A user can become a new member of a role as long as the number of members allowed for the role is not exceeded in particular subject.

6 ARBAC97

ARBAC97 consists of three sub-models:

- URA97- for managing user-role assignment,
- PRA97 - for managing permission-role assignment
- RRA97 for managing role-role assignment.

6.1 THE URA97 MODEL FOR USER-ROLE ASSIGNMENT

The URA97 model used for managing user-role assignment. The URA97 have two steps:

1. Granting a user membership in a role
2. Revoking a user's membership.

URA97 is quite powerful and goes much beyond existing administrative models for user-role assignment. It is also applicable beyond RBAC to user-group assignment.

6.2 THE PRA97 MODEL FOR MANAGING PERMISSION-ROLE ASSIGNMENT

PRA97 is concerned with role-permission assignment and revocation. They are essentially entities that are brought together by a role. Hence, PRA97 to be a dual of URA97. The notion of a prerequisite condition is identical to that in URA97, except the Boolean expression is evaluated for membership and non membership of permission in specified roles.

6.2 THE PRA97 MODEL FOR MANAGING ROLE-ROLE ASSIGNMENT

The RRA97 model for role-role assignment perform the abilities, Groups, and UP-Roles

6.2.1 Abilities, Groups, and UP-Roles

For role-role assignment, we distinguish three kinds of roles

- Abilities are roles that can only have permissions and other abilities as members.
- Groups are roles that can only have users and other groups as members.
- UP-Roles are roles that have no restriction on membership, i.e., their membership can include users, permissions, groups, abilities, and other UP-roles.

7 ORACLE

The Oracle DBMS implements the notion of roles since early 1990s, and it includes support for administration of the access control state. Unlike ARBAC, Oracle's RBAC administrations have been widely used in real world; Oracle thus presents an invaluable reality check for administrative approaches to RBAC. The success of RBAC research is partially due to the fact that the notion of roles has been implemented in commercial systems, so that the research can be guided by real-world experiences [17]. We believe research on administrative models for RBAC must also learn from existing systems such as Oracle. There are two kinds of privileges in Oracle: system privileges and object privileges. There are over 100 system privileges in Oracle 10g. For example, the "create role" system privilege allows one to create a new role, "drop any role" allows to drop any role, "grant any role allows to grant any role to a user or another role". An object privilege identifies an object, which is either a table or a view, and an access mode, which is one of the following: select, insert, update and delete. Oracle's permission management is a hybrid of DAC (Discretionary Access Control) and RBAC. Privileges can be granted to users and to roles. And roles can be granted to roles and to users. A system privilege or a role can be granted "with admin option". If a user is granted a role with admin option, then we say the user has admin power over the role. This enables the user to grant the role to other users and roles as well as to revoke the role from other users or roles. A role r1 can also be granted to another role r2 with admin option, in which case any user that is a member of r2 has admin power over r1. A user can create a role if he has the create role system privilege and the role to be created does not already exist. When a role is created, the creator will be automatically granted the role with admin option. This enables the creator to further grant the role to any other role or user.In Oracle, if one has control over permission, then one can grant the permission to any role; no control over the role is needed. This is different from the approach in ARBAC97 and SARBAC, in which granting a permission to a role is viewed as a dual of assigning a user to a role, and requires the granter has some kind of control over the role. Oracle's design seems more intuitive. Granting a role to a user implies giving out privileges associated with the role; thus some control over the role is needed. Similarly, granting a permission to a role implies giving out the permission; thus some control over the permission (rather than over the role) is needed. On the other hand, Oracle's approach leads to a denial of service attack: Any user who has the "create role" system privilege can stop other users from logging in. When a user logs in, a set of roles that the user has are activated, as is any role that has been granted to one of these roles. Oracle has a limit on the number of roles that can be activated in a session; if a user has more roles, then the user cannot log in. Oracle has a predefined role called PUBLIC, which is granted to every user and is activated by default. Any user who has the "create role" system privilege can create a large number of roles and grant them to PUBLIC, resulting in other users unable to log in.

8 CONCLUSION

The RBAC System helps to manage the organization. For the user, roles and Permissions are given to access their respective functions. The user permission is given based on their responsibilities and the job entitlement. RBAC system will help to manage and which brings the enterpriser to communicate and assign roles and responsibilities each time when they change or evolve in an work assignment.

9. ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their helpful comments. We thank our Head of the Department and our Students .We also thanks our family members for their support.

Volume : 3 | Issue : 3 | March 2013 | ISSN - 2249-555X

REFERENCE [1] J. Becker, M. Rosemann, C. von Uthmann, Guidelines of business process modeling, in: Business Process Management, Models, Techniques, and Empirical Studies, Lecture Notes in Computer Science (LNCS), vol. 1106, Springer Verlag, 2000. | [2] C. Ouyang, M. Dumas, W.M.P. van der Aalst, A.H.M. ter Hofstede, J. Mendling,From business process models to process-oriented software systems, ACM | Transactions on Software Engineering and Methodology (TOSEM) 4 (1) (2009). | [3] E.A. Stohr, J.L. Zhao, Workflow automation: overview and research issues, Information Systems Frontiers 3 (3) (2001). | [4] S. Oh and R. S. Sandhu. A model for role administration using organization structure. In Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies (SACMAT 2002), June 2002. | [5] R. S. Sandhu and V. Bhamidipati. Role-based administration of user-role assignment: The URA97 model and its Oracle implementation. Journal of Computer Security, 7, 499. | [6] R. S. Sandhu and Q. Munawer. The ARBAC99 model for administration of roles. In Proceedings of the 14th Annual | Computer Security Applications Conference, | [7] D. F. Ferraiolo, R. Chandramouli, G.-J. Ahn, and S. Gavrila. The role control center: Features and case studies. In Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies, June 2003. | [8] ANSI. American national standard for information technology – role based access control. ANSI INCITS | 359-2004, Feb. 2004. | [9] D. F. Ferraiolo, R. S. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed NIST standard for role-based access control. ACM Transactions on Information and | Systems Security, 4(3):174–274, Aug. 2001. | [10] A. Kern, A. Schaad, and J. Moffett. An administration concept for the enterprise role-based access control model. In | Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies (SACMAT 2003), pages 3–11, June 2003. | [11] N. Li, W. H. Winsborough, and J. C. Mitchell. Distributed credential chain discovery in trust management: extended abstract. In Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS), 2001 | [13] R. L. Rivest and B. Lampson. Simple distributed security infrastructure. Presented at CRYPTO'96 Rump session, 1996. | [14] M. Nyanchama and S. Osborn. The role graph model and conflict of interest. ACM Transactions on Information and System Security, 2(1):3–33, Feb. 1999. | [14] M. Nyanchama and S. Osborn. The role graph model and conflict of interest. ACM Transactions on Information and System Security, 2(1):3–33, Feb. 1999. | [15] L. Giuri and P. Iglio. Role templates for content-based access control. In Proceedings of the Second ACM Workshop on Role-Based Access Control (RBAC'97), pages 153–159, Nov. 1997. | [16] R. S. Sandhu and V. Bhamidipati. Role-based administration of user-role assignment: The URA97 model and its Oracle implementation. Journal of Computer Security, 7, 1999. | [17] R. S. Sandhu, V. Bhamidipati, E. Coyne, S. Ganta, and C. Youman. The ARBAC97 model for role-based administration of roles: preliminary description and outline. In Proceedings of the Second ACM workshop on Role-based | access control (RBAC 1997), pages 41–50, Nov. 1997. | [18] R. S. Sandhu, V. Bhamidipati, and Q. Munawer. The ARBAC97 model for role-based aministration of roles. ACM Transactions on Information and Systems Security, 2(1):105–135, Feb. 1999. | [19] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. IEEE Computer, | 29(2):38–47, February 1996. | [20] R. S. Sandhu and Q. Munawer. The ARBAC99 model for administration of roles. In Proceedings of the 18th Annual Computer Security Applications Conference, pages 179–188,Dec. 1999. |