AN EXPERIMENTAL STUDY OF ERBAC03 FOR ACCESS CONTROL ADMINISTRATION

Burin Yenmunkong and Chanboon Sathitwiriyawong

Faculty of Information Technology, and
Research Center for Communications and Information Technology (ReCCIT)
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
Email: bom_burin@hotmail.com, chanboon@it.kmitl.ac.th

ABSTRACT

This paper proposes an extension of the conventional role-based access control (RBAC model) called enhanced role-based access control or ERBAC03 model. The model is developed for the role-based access control of information system resources in large organizations that have many branches. Each branch consists of many users with different roles. It is related to a specified static separation of duty constraint in order to prevent fraud of users. We analyze the result of RBAC and ERBAC03 models by a number of experiments based on users' locations. The result proves that the proposed ERBAC03 model eliminates the inaccuracy of access control administration that has incurred in the conventional RBAC model. Any conflicting role is not allowed to be added to the associated database tables.

1. INTRODUCTION

Role-based access control has recently received considerable attention as a promising alternative to traditional discretionary and mandatory access controls [1]. In RBAC96 model [2], permissions are associated with roles and users are members of appropriate roles. Therefore, roles permissions can simplify the management of permissions.

An enhanced role-based access control (ERBAC03) [3] was recently developed to support an organization that has many branches. Each branch consists of many users with different roles. This model can prevent the security violation of users from acquiring their roles at different locations. The management of a centralized system is easier than a distributed system by a small group of security administrator in a large organization environment. The static separation of duty concept can be applied to the ERBAC03 model [4] to prevent fraud of users in the organization.

Botha and Perelson [5,6] proposed a mathematical model based on the concept of "conflicting entities" to express static separation of duty requirements. The ERBAC03 model also applies this concept to gain higher efficiency of the access control technology.

The design and development of ERBAC03 model led to the hypothesis of experiment based on NIST [7]. This paper demonstrates the main algorithm by defining non-conflicting roles to users. The experimentation compares the result of the proposed ERBAC03 model with the existing RBAC model which was analyzed and summarized to provide the highest efficiency in large organization environment.

2. BASIC CONCEPT OF ACCESS CONTROLS

This topic explains the concept of the conventional role-based access control (RBAC) and the proposed enhanced role-based access control (ERBAC03) models, including the static separation of duty (SSoD) constraints for the ERBAC03 model.

2.1. RBAC Model

The concept of roles is pivotal in the role-based access control. Users can be anyone or anything who receives access permissions on information system resources based on their roles. Roles often correspond to positions in the organization structure. Permissions can be interpreted as the right to execute a certain method of an object.

Roles may be related to a partial order. A role inherits permissions assigned to the roles that are the junior to it in the partial order. For example, the 'Chief Post Office' role may be considered as the senior to the 'Accountant' role. Therefore, the 'Chief Post Office' role will inherit the permissions assigned to the 'Accountant' role.

2.2. ERBAC03 Model

In our previous work [3], we extended RBAC model by defining three additional entities as shown in figure 2. The concept of the ERBAC03 model was influenced by the concept of Epstien [8].

The ERBAC03 model, as shown in figure 1, requires the modification of the endpoints (i.e., roles and permissions) by introducing three additional entities that consist of locations, jobs and tasks. We concentrate on the creation of the elements between the endpoints, such as jobs and tasks, in order to

0-7803-8560-8/04/$20.00©2004IEEE

make the permissions to roles more clear. It can also be applied to large enterprises effectively.

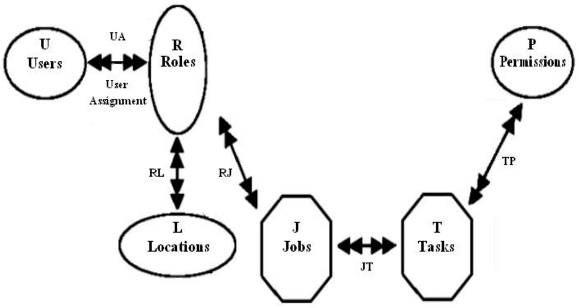

Figure 1. ERBAC03 model

We consider that a location comprises many roles. Each location is related to roles and reflects large enterprises that have many branches. Each branch consists of many roles that reflect the model management in the distributed aspect. A role may perform more than one type of work. A role is responsible for all the activities that are required to perform the work. We define each type of work as a job. The job does not need to be in any sequence. We also show that the tasks requiring access to applications will be mapped to the permissions granted the desired access. The difference in components between RBAC model and ERBAC03 model is summarized in table 1.

Table 1. Components comparison between RBAC model and ERBAC03 model

RBAC Model	ERBAC03 Model
Users	Users
Roles	Roles
	Locations
	Jobs
	Tasks
Permissions	Permissions

3. CONSTRAINT FEATURES OF ERBAC03 MODEL

Static separation of duty (SSoD) constraints are used in ERBAC03 model for the prevention of fraud to ensure that a single user does not have too much authority. The authority is set as a permission based on locations. Therefore, the essence of our paradigm depends on the conflicting permissions and locations.

Conflicting users (CU) are users who are likely to conspire. In practice, this may be family members or previously known accomplices. Conflicting users are sometimes considered as a single user.

Conflicting roles (CR) are roles that together have the ability to conspire, i.e. they are assigned some conflicting permissions. Consider, for example, 'Chief Post Office' role and 'Accountant' role. It is a common practice that the chief post office and the accountant should be independent. Roles may have certain permissions in common, e.g. 'View Financial Table' permission, whilst 'Edit Financial Table' permission and 'Audit Financial Table' permission may be assigned only to one of these roles.

Conflicting locations (CL) are locations that require conflicting roles to complete. For example, 'Bangkok' location and 'Krabi' location would be conflicting since they require both 'Chief Post Office' role and 'Accountant' role.

Conflicting jobs (CJ) are jobs that together have the ability to cause frauds. For example, 'Close End of Day Account' job and 'Mail Issuer' job would be conflicting since they require both 'Count Money' task and 'Approve Financial' task.

Conflicting tasks (CT) are tasks that require conflicting permissions to complete. This would, for example, imply that 'Count Money' task and 'Approve Financial' task would be conflicting since they require both 'Read Financial Record' permission and 'Write Financial Record' permission. These permissions are, in turn, conflicting.

Conflicting permissions (CP) are permissions that can result in unnecessary power if bestowed on the same person. For example, a person with a permission required for financial audits should not acquire a permission to approve financial transactions. If this were allowed, auditors could lose their independence.

4. ALGORITHMS OF ERBAC03 MODEL

The ERBAC03 model outlines algorithms for each of the possible actions that may be performed upon the entities. They are designed according to the specifications that have been defined in our previous work [4].

The relational database system terminology is used to describe the proposed ERBAC03 algorithms. A conceptual entity-relationship diagram for the ERBAC03 model is defined as shown in figure 2.

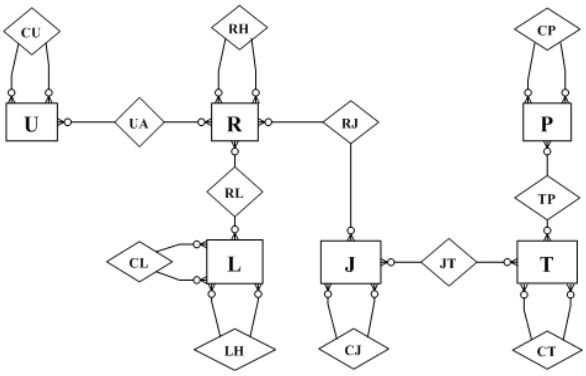

Figure 2. Conceptual entity-relationship diagram

The associations are depicted by the UA, RL, RJ, JT, TP, LH, and RH database tables. Each association allows a many-to-many relationship between one entity and another entity. The LH (location hierarchy) and RH (role hierarchy) association is slightly different from the others. It is used to create both location network and role network.

The conflicts between entities are depicted by the CU, CR, CL, CJ, CT, and CP database tables. As can be seen, a conflict is a many-to-many relationship between the entities of a particular set.

Finally, there are database tables that represent the sets of entities. They are denoted as U, R, L, J, T, and P entities, respectively.

Within each category, there exists the potential of various conflicts. Every conflict needs an algorithm that outlines what an administration tool will do to test whether the action will violate the integrity requirement and will not be allowed to continue.

The main algorithm of ERBAC03 model can be divided into six steps as shown in figure 3. It includes the association action and the creation of entities. It is assumed that a set of entities and their conflicts already exist within the administration environment.

Step 1: Does user u_i have any conflicting users? This is done by an iteration process through all the conflicting user records where one of the users in the record is the user being added to the association. If the answer is yes, proceeding to step 2, otherwise proceeding to step 4.

Step 2: Search for any roles that have already been associated to the conflicting users found in step 1. This is done by an iteration process through the entire user to role association table where any of the associated users are in the group of users found in step 1. If no roles are found, the association can be safely made, otherwise proceeding to step 3.

Step 3: Do any of the roles found in step 2 conflict the added role r? This is accomplished by an iteration process through the conflicting roles records where the current role and any of the roles found in step 2 form a record. If none are found, the association can be safely made, otherwise the association may not be proceeded.

Step 4: Search for any roles that have already been associated with non-conflicting users found in step 1. This is done by an iteration process through the entire user to role association table where any of the associated users are in the group of users found in step 1. If no roles are found, the association can be safely made, otherwise proceeding to step 5.

Step 5: Do any of the roles found in step 4 conflict the added role r? This is accomplished by an iteration process through the conflicting roles records where the current role and any of the roles found in step 4 form a record. If none are found, the association can be safely made, otherwise the association may not be proceeded.

Step 6: Add the association as a record to the appropriate database table.

The above algorithm describes exactly how the ERBAC03 model maintains the integrity of the access control data in the associated database tables.

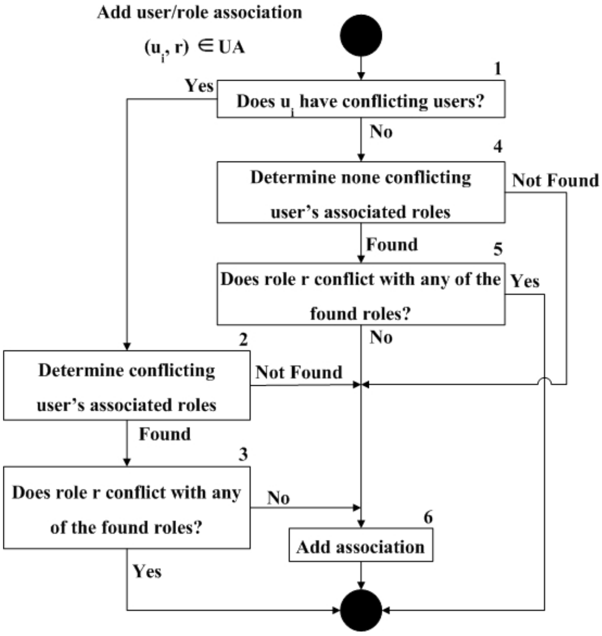
Add user/role association
$(u_i, r) \in UA$
1
Yes
Does u_i have conflicting users?
No
4
Determine none conflicting user's associated roles
Not Found
Found
5
Does role r conflict with any of the found roles?
Yes
No
2
Determine conflicting user's associated roles
Not Found
Found
3
Does role r conflict with any of the found roles?
No
6
Add association
Yes

Figure 3. Adding user/role association

5. EXPERIMENTATION OF ERBAC03 MODEL

We developed a hypothesis based on the benefit of ERBAC03 model [7]. Table 2 lists the hypothesis that can impact security violations from the inaccuracy of access control administration.

Table 2. A hypothesis and metric of experimentation

Hypothesis	ERBAC03 can eliminate erroneous access control administration of RBAC by specifying static separation of duty constraints.
Metric	Elimination of inaccurate access control administration

5.1. Experimental design

The experimental data owned by Thailand Post Company is the login data into the computer system that was collected in the controlled environment comprising three departments as shown in table 3. The collection of data was done by the Oracle database feature called "Audit". The conventional RBAC technology is used in the company.

Table 3. An example data of the RBAC model in Thailand Post Company

USERID	ROLEID	TERMINAL
Burin	ROAPRD	WRKDBA_01
Administrator	ROAPRD	WRKCDES_03
SYSTEM	ROAPRD	WRKCSMS_02
Zintoo	ROAPRD	ZINTOOXP
Anan	ROAPRD	WRKDBA_02

In table 3, we observe a "ROAPRD" role as an administration role. Only database department can authorize this role. Terminals "WRKDBA_01" and "WRKDBA_02" are located in this department. Another terminal is unauthorized on the "ROAPRD" role. The experimental data shows the attack of users who attempt to login to the computer system by the "ROAPRD" role that a denial of service problem may occur. The percentage of data accuracy can be computed as follows.

$$\text{Data accuracy }(\%) = \frac{\text{Number of accurate data}}{\text{Number of measured data}} \times 100$$

An experiment was designed to compare the administration accuracy between the conventional RBAC model and the proposed ERBAC03 model. The ERBAC03 model uses the algorithm shown in the previous section. We use this algorithm to develop database triggers [9] to enforce the conflict paradigm constraints. It is the primary focus of this experimental study.

5.2. Experimental Result

The experimental data of the RBAC model during one-week period has 4,244 times of logins into a computer system. Only 270 times of logins are accurate. All system connections can be accessed from 26 different locations. The result of data accuracy is as low as 6.36%.

The experimental result shows that the inaccurate access control administration that occurs during the one-week period is 93.64 percent. This may cause system down. However, after applying the ERBAC03 model in the computer system, the number of login time is 353 times whereas all of the correct data have the same value. The result shows that the proposed ERBAC03 model can eliminate all security violations. Therefore, it can control the access of users in the computer system more effectively than the conventional RBAC model.

6. CONCLUSION

This paper proposes an extended RBAC model called "Enhanced Role-Based Access Control (ERBAC03)". The new model requires the modification of the end-points (roles and permissions) by introducing three additional entities which consist of locations, jobs, and tasks. This model relates to the specified static separation of duty constraints to prevent fraud of users in the organization. We give the definition of conflicting entities to develop the algorithm of user to role association used in the experimentation. The result shows that ERBAC03 model can gain higher efficiency than RBAC model for a large organization environment comprising many branches.

Since the association of any conflicting role is not allowed to be added to the target database table, all of the data in the user-to-role table are always correct. Finally, ERBAC03 model can reduce the denial of service problem in information systems effectively.

7. REFERENCES

[1] S. Osborn, and R. Sandhu, "Configuring Role-based Access Control to Enforce Mandatory and Discretionary Access Control Policies," *ACM Transactions on Information and System Security (TISSEC), Volume 3, Issue 2*, pp. 85-106, May 2000.

[2] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role Based Access Control Models," *IEEE Computer, Volume 29, Issue 2*, pp. 38-47, February 1996.

[3] C. Sathitwiriyawong, and B. Yenmunkong, "An Enhanced Role-Based Access Control for Large Enterprises," *Proceedings of the 3rd International Symposium on Communications and Information Technologies (ISCIT 2003)*, pp. 279-283, September 2003.

[4] B. Yenmunkong, and C. Sathitwiriyawong, "An Enhanced Role-Based Access Control Model using Constraint Features," *Proceedings of the 7th National Computer Science and Engineering Conference (NCSEC 2003)*, pp. 103-108, October 2003.

[5] R. A. Botha, and J. H. P. Eloff, "Separation of Duties for Access Control Enforcement in Workflow Environments," *IBM Systems Journal, Volume 40, Number 3*, 2001.

[6] S. Perelson, and R. A. Botha, "Conflict Analysis as a Means of Enforcing Static Separation of Duty Requirements in Workflow Environments," *South African Computer Journal, Volume 26*, pp. 212-216, November 2000.

[7] Gregory T. "The Economic Impact of Role-Based Access Control," *NIST Planning Report 02-1*, March 2002.

[8] P. Epstien, and R. Sandhu, "Engineering of Role/Permission Assignments," *Proceedings of the 17th Annual Computer Security Applications Conference*, pp. 127-136, December 2001.

[9] Oracle, *Application Developer's Guide Fundamentals [Online], Available: http://www.oracle.com*, Oracle Corporation, 1999.