CHAPTER 32

An Integration Model of Role-Based Access Control and Activity-Based Access Control Using Task

Sejong Oh, Soeg Park
Sogang University

Key words: Access control, RBAC, Task, Role, Enterprise environment

Abstract: Role-based access control (RBAC) and activity-based access control (ABAC) models are well known and recognized as a good security model for enterprise environment. (ABAC model is represented as 'workflow'). But these models have some limitations to apply to enterprise environment. Furthermore, enterprise environment needs application both RBAC and ABAC models.

In this paper we propose integration model of RABC and ABAC. For this we describe basic concept and limitations of RBAC and ABAC models. And we introduce concept of classifications for tasks. We use task by means of connection RBAC and ABAC models. Also we discuss the effect of new integration model.

1. INTRODUCTION

In general, today's companies manage business information with computer systems. Access control is an important security issue in the enterprise environment. Access means the ability to perform work such as reading, writing, and the execution of the system resources. Access control is the way to control the ability for performing the work.

From an access control point of view, enterprise environment can be expressed in Figure 1. In general, users in the company belong to the organization structure and they are performing their assigned job functions according to their job positions. Organization structure reflects authorization structure. Users read or write information resources for executing their job functions. There are two ways that users access information resources. First,

users can access information resources directly for their some job functions. Second, some job functions are connected with others in the business process, and direct access of information resources is restricted by the status of business process. Passive access control applies to the first case, and active access control applies to the second case.

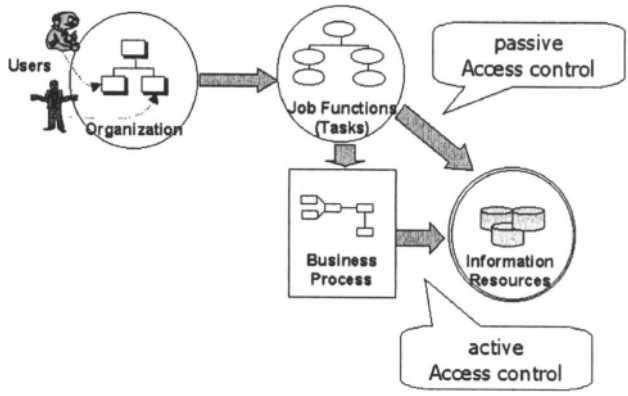

Figure 1. *Enterprise Environment*

Note. *We will use the term* ***'task'*** *instead of* ***'job function'.*** *In this paper, task has a meaning of unit of job that accesses information resources.*

Researchers have developed some access control models such as discretionary access control (DAC)[1], mandatory access control (MAC)[2], and role-based access control (RBAC). Activity-based access control (ABAC) model, which was motivated by workflow environment, was introduced recently. RBAC and ABAC are more suitable for enterprise environment than MAC and DAC. RBAC and ABAC have a good point of security, but also have constraints about applications for enterprise environment. So it is necessary to investigate more proper model for enterprise environment.

The purpose of this paper is to propose an improved access control model for enterprise environment through the integration of the RBAC and ABAC models. At first we reviews the limitations of RBAC and ABAC models, and then introduces our improved access control model.

2. RBAC AND ABAC MODELS

Role-based access control (RBAC)[5][6] has the central notion of preventing users from accessing company information discretionarily. Instead, access rights are associated with roles, and users are assigned to appropriate roles.

RBAC model has limitations as follows.

– RBAC supports passive access control. For example, if an access right is assigned to a user, then he/she can use the access right at any time. But enterprise environment include workflow, and it needs a dynamic activation of access right. RBAC cannot support dynamic activation of access right.
– Basic RBAC model has a role hierarchy concept that higher role inherits all access rights of lower role in the role hierarchy, and it is not suitable for real world. For example, *manager* is a higher job position than that of *clerk,* however, manager doesn't automatically inherit the '*register purchase'* job function of *clerk.* Full inheritance of role hierarchy has undesirable side effects by violating 'need-to-do' principle.

Activity-based access control (ABAC) model [3] [8][9] is investigated for a collaborative work environment represented as 'workflow'. Workflow is defined as a set of activities (tasks) that are connected to achieve a common goal. ABAC separates access right assignment for users and access right activation. Even if a user was allocated access rights on the workflow template, he/she can exercise his rights during the activation of the task in the specific workflow instance. ABAC model has limitations in the enterprise environment as follows.

– There exist many tasks that don't belong to workflow in the company, and ABAC model doesn't deal with them. So extra access control methods should be added to ABAC model.
– In the real world, a superior officer supervises and reviews execution of tasks of his/her inferior clerks. It's important for security and integrity; however, ABAC model doesn't take review and supervision into consideration.

3. INTEGRATION MODEL OF RBAC & ABAC

3.1 Problems in the Integration of RBAC & ABAC Models

As we can see, enterprise environment needs both passive and active access controls. (See Figure 1). We choose RBAC for passive access control model. Some researchers proposed injection of RBAC to workflow security [4] [7], But their approach doesn't deal with RBAC and ABAC models on an equal footing. Their approach is based on ABAC model, and adopts concept of 'role' as a meaning of group. There are some problems in the integration of RBAC and ABAC models as follows.

First, task is a unit of permission in the ABAC model. But RBAC, as a passive access control model, assigns information objects such as file or

record to role. Task is higher level than information object. Integration model needs consistent unit of permission between RBAC and ABAC models.

Second, as we pointed out in section 3, there exist many tasks that don't belong to workflow in the company. In this case passive access control is more proper than active access control.

Third, as we pointed out in section 2, full inheritance of role hierarchy in RBAC has undesirable side effects. These side effects bring about serious security problems in active access control such as domination of 'need to do' principle.

3.2 Task Classification Concept

Before we propose integration model RBAC and ABAC that solves above problems, we introduce task classification concept. We will use a task concept as a connecter of RBAC and ABAC model.

By observation of the enterprise environment, we found that there are three classes of tasks such as in Table 1. If a user U_1 has tasks that belong to class S, their related access rights are inherited to user Un who has a higher job position than U_1 in the organization structure. But class W and class P do not have such inheritance characteristics. Tasks belong to class W, which has a relation with workflow and show the characteristics of an ABAC model. Passive security model is applied to class S and class P. Access control of the enterprise environment needs a proper method to deal with three classes of tasks through different ways. Our suggested model is based on the classification of tasks.

***Table 1**. Classification of tasks.*

Classification of tasks		Class id	Example tasks	Characteristics	Inherited to higher job positions	Applied security model
Supervision Tasks		Class S	✔ supervise ✔ review ✔ delegation	✔ has a access right inheritance ✔ access right hierarchy is similar to organization hierarchy	O	Passive
Essential tasks	Workflow oriented tasks	Class W	✔ *drafting &* approval ✔ chained job	✔ similar to transaction ✔ needs active access control ✔ always includes write operation	X	Active
	Non workflow oriented tasks	Class P	✔ *analysis* ✔ planning ✔ decision making	✔ private tasks (has no relationship with other tasks or other job positions)	X	Passive

3.3 Integration of RBAC and ABAC

Now we propose the integration model of RBAC and ABAC models based on task classification. *(Note. We will call new integration model as T-RBAC. It means that Task-Role-Based Access Control).* Figure 2 shows a brief of T-RBAC. The most difference between T-RBAC and RBAC is that the access rights are assigned to task in T-RBAC, rather than access rights are assigned to role in RBAC. In the real world access rights are needed for the user to perform tasks. So assignment of access rights to task is reasonable. Another difference is role hierarchy. We use supervision role hierarchy (S-RH) instead of general role hierarchy. In the S-RH, higher role doesn't inherit all access rights of lower role in the role hierarchy. Only access rights of class S are inherited from lower role to higher role.

Tasks in the class W are used to compose workflow. Workflow creates the workflow instances that are set of task instances. Access rights are assigned to tasks in the class W statically. But the access rights are bound and activated during execution of task instance. Task instance has three attributes such as activation condition, time constraint, and cardinality. Time constraint is an available time after the task is activated. Cardinality is the number of specific task instance at the same time. How to specify and manage security constraint is remained research issue.

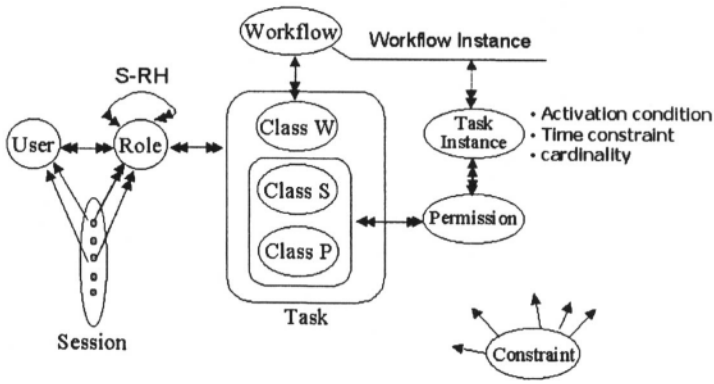

***Figure 2. T-RBAC** : Integration model of RABC and ABAC models*

The step of authority check in T-RBAC is as follows. When a user request accessing to some information objects, RBAC system checks the validity of the user's role, task. If the user's role and task are correct, then RBAC system checks the validity of permission. If the task belongs to class S or class P, RBAC system checks that permission is assigned to the task or not. If the task belongs to class W, RBAC system specifies task instance and checks activation condition, time constraint, and cardinality of the task instance. After checking permission, RBAC system checks security and

integrity constraints. And RBAC system decides to accept or reject user's request.

In T-RBAC model, the concept of session and user-role assignment (URA) follows RBAC.

4. CONCLUSION

There are two central ideas in the T-RBAC. One is the classification of enterprise tasks (job functions) according to their characteristics. The other is to use intermediate tasks between access rights and roles instead of assigning access rights to roles. It makes possible that roles can be linked to access rights through intermediate tasks. Moreover, it makes the point of contact that RBAC could be integrated to ABAC model. The T-RBAC model has following effect from their characteristics.

– T-RBAC can support more elaborate access control. In the RBAC model, the unit of separation of duty and delegation is a role unlike in the T-RBAC where the unit is task. Task unit has more small scope of access rights than role unit
– It offers the criterion that which task/access rights can be inherited to higher roles from lower roles on the supervision role hierarchy (S-RH). Only the tasks belong to class S has an inheritance characteristic. It solves problems of general role hierarchy in RBAC.
– T-RBAC deals each class by different way according to its class. It is also possible to apply *active security model* to tasks that belong to class W and apply the general *passive security model* to tasks that belong to class S or P. Thus, task is a base concept for the integration of RBAC and ABAC.

REFERENCES

[1] C.P.Pfleeger, Security in Computing, second edition, Prentice-Hall International Inc.,1997.
[2] E.G.Amoroso, Fundamentals of Computer Security Technology, PTR Prentice Hall, 1994, 253-257.
[3] Dagstull, G.Coulouris, and J.Dollimore, "A Security Model for Cooperative work : a model and its system implications", Position paper for ACM European SIGOPS Workshop, September 1994.
[4] G.J.Ahn, R.S.Sandhu, M.Kang, and J.Park, "Injecting RBAC to Secure a Web-based Workflow System", Proc. of 5th ACM Workshop on Role-Based Access Control. 2000.
[5] R.S.Sandhu, E.J.Coyne, H.L.Feinstein, and C.E.Youman, "Role-Based Access Control Method", IEEE Computer, vol.29, Feb. 1996.
[6] D.Ferraio, J.Cugini, and R.Kuhn, "Role-based Access Control (RBAC): Features and motivations", Proc. of 11th Annual Computer Security Application Conference, 1995.12.
[7] W.K.Huang and V.Atluri, "SecureFlow: A Secure Web-enabled Workflow Management System", Proc. of 4th ACM Workshop on Role-Based Access Control, 1999.
[8] G.Herrmann and G.Pernul, "Towards Security Semantics in Workflow Management", Proc. of the 31st Hawaii International Conference on System Sciences, 1998.
[9] R.K.Thomas and R.S.Sandhu, "Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management", Proc. of the IFIP WG11.3 Workshop on Database Security, 1997.