Information Security based on Resource Conflict with SSH Architecture

Su Yu
College of Electronic and Electrical Engineering
Shanghai University of Engineering Science
Shanghai, China
suyu_sh@hotmail.com

Yin Wang
Department of Mathematics and Computer Science
Lawrence Technological University
Southfield, MI, 48075, USA
ywang12@ltu.edu

Ghassan M. Azar
Department of Mathematics and Computer Science
Lawrence Technological University
Southfield, MI, 48075, USA
gazar@ltu.edu

Zhou Wei
College of Electronic and Electrical Engineering
Shanghai University of Engineering Science
Shanghai, China
zhouwei@sues.edu.cn

***Abstract*—Role-based access control has become the predominant model for advanced access control. SOD of RBAC is a good way to solve the problem of resource allocation and right management for improving the security of the system. According to the needs of the MVC-based information management system security, this paper researches separation of duties (SOD) in Role-Based Access Control, and then gives a program to implement RBAC model as the framework of SSH. Finally, this paper describes how to realize the RBAC in a specific information management system.**

keywordss*—- *RBAC, SOD, SSH, MVC, Security Constraint

I. INTRODUCTION

Due to the popularity of the Internet and the development of network technology, most of the enterprises or units have their own Web sites. Through the Internet or an intranet, enterprise management has become more convenient; enterprise information dissemination has become more convenient; enterprise marketing has become much easier. The most number of Web systems exist to this problem: the function is powerful, but there are a lot of security risks. Different types of users in the system have the same privileges to some information, and they can modify and delete arbitrarily, which is very dangerous for some important information. As scientific management of information, the highest level the user has the highest authority of the information, and at the bottom or sub underlying user has the least permissions information. In addition, most of the Web system are the lack of a good access control mechanisms. In the mechanism users, roles and privileges are set, and then the roles are assigned for different users, the privileges are assigned for different roles. Finally privileges are associated with the users, for formation of an effective system of safety management. In conclusion, when the enterprises build their own web-based management system, not only to consider the functional integrity and simplicity of operation, but also to consider the security of the system. Under the promise of safety, the functional integrity and simplicity of operation makes sense.

Compared with traditional lattice-based access control policies, such as Discretionary Access Control (DAC) and Mandatory Access Control (MAC) developed primarily for military systems(see Sandhu [2] for a discussion of these), RBAC can more effectively meet the needs of commercial systems[3][4]. Role-based access control mechanisms rely on convenient resource management. In RBAC there are two important factors. They are role inheritance and separation of duties. Role inherited makes resource allocation management easier; Role mutex achieves separation of duties and improves efficiency and stability of system. Separation of duties has some theoretical results, and many researchers have given a variety of theoretical models. But in the previous literature, how to implement the separation of duties in the information management system was rarely described. The implementation of the separation of duties is far less than the theoretical research. So, the implementation of the separation of duties still need further study. [1]

The main goal of this paper studies RBAC model structure, specially focus on SOD and restrictions. After intensive research RBAC theory, this paper designs the RBAC model structure based on SOD and it can be used in company's information management systems. The RBAC model structure developed by this project can be used in the web-based information management system. The system uses SSH framework based on MVC, development software is Java, database is MYSQL. Access control software based on RBAC can be applied to the the waigaoqiao company's management system.

II. SEPARATION OF DUTY (SOD)

Separation of duties (SOD) is the concept of having more than one person required to complete a task. For example, in business one person can not be an accountant while he is a cashier. RBAC requires separation of duty to prevent a role from having too many permissions. By this, RBAC can secure system. Through role conflict, we can achieve SOD. Role assignment is an important part of RBAC. But when apply RBAC into a real system, security problems occurs because of no constraint on role assignment. So SOD is necessary to RBAC model. System security partly relies on SOD. What SOD does in RBAC is to separate roles which conflicts with each other. Strong constraint does not allow any roles being added into role set. Weak constraint does not allow roles in a special event being added into role set. Combining with other limits, such as frequency constraint and cardinality constraint, user will get through an SOD firewall before role assignment, separating conflicting roles, making role assignment and resources fetching safely.

SOD is very important in business system, also is one of the most desired features in RBAC system. Administration constraints may need to be enforced to prevent information misuse and prevent fraudulent activities. A typical authorization constraint, broadly relevant and well recognized, is separation of duties (SOD). Reducing the risk of fraud by not allowing any individual to have sufficient authority within the system to single-handedly perpetrate fraud is the intent of SOD. Such constraints can be easily expressed using an RBAC model through SOD constraints on roles, user-role assignments, and role-permission assignments. Furthermore, using constraints on the activation of user assigned roles, users can sign on with the least privilege set required for any access. In case of inadvertent errors, such least privilege assignments can contain damage.

As shown in Figure 1, it consists of three parts: users, roles and permissions. The three roles (administrator, accountant, clerk) are defined to be mutually exclusive; administrator has permission to sign the checks, the accountant has permission to prepare checks, the clerk has permission to deliver checks; three users, Jack, Tom and Rose, respectively to be assigned to the roles of administrator, accountant and clerk; the initial state of system is static separation of duties. A user can entrust own role to other users. When Rose's role is actived he can delegate the clerk' role to Tom. So Tom has two mutually exclusive roles, accountant and clerk. But he only can temporarily have a role, and he can not activate two roles at the same time. The state of the system is changed from static separation of duties to the dynamic separation of duty. If the clerk role and accountant role have permissions to the operation of the same check, the system will enter the state of object-based static SOD. It is that user assigned to clerk role and accountant role can not have operation on the same check more than two.

If Jack also delegates his role to Tom, then Tom would have three roles: administrator, accountant and clerk. Because these three roles can complete the entire check task it is contrary to the role based on the operation-based static SOD. Users can be assigned conflict roles, but not be able to have all operations for completing a task. If Tom activates three mutually exclusive roles at the same time without same check (that is, the same object), the system will enter the state of history dynamic SOD. SOD constraint strength shown in Figure 2, from low to high, from strong to weak.

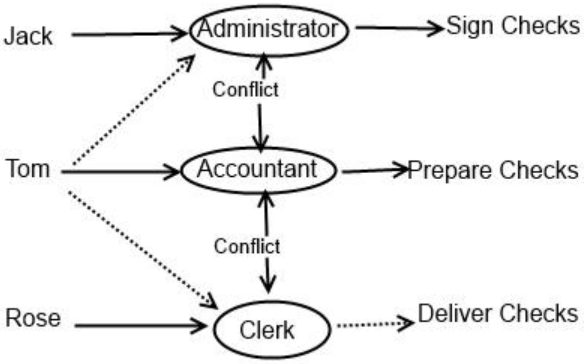

Figure 1. Separation of Duties

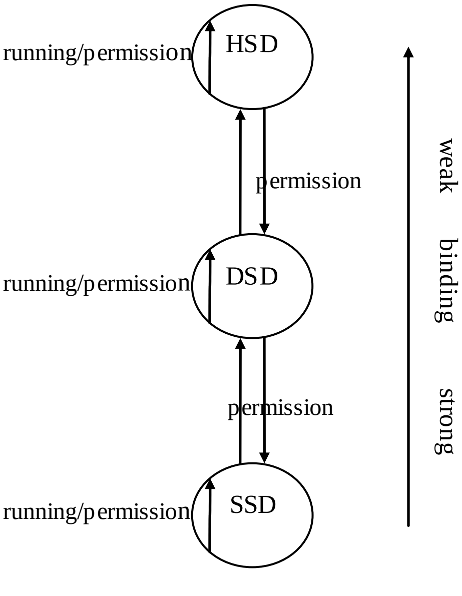

Figure 2 SOD constraint relationship

The strongest constraint is in the state diagram at the bottom. While implementing the strategy of SOD, the weak constraints of SOD is more flexible and more easy to implement separation of duties, the strong constraints is more difficult to achieve due to too strict. DSD constraint is mainly carried out in the event. For example, an user need enter an event, such as the number of roles that user is assigned is greater than or equal two, the DSD constraint will be droved, if the role of user is in DSD, the role is a constraint role and then user can’t be enter . When user is assigned a role, SSOD constraint well be judged. In case of its own role and selected role in the SSD, the role is a constraint role and role should be assigned selected role.

III. SSH WORK MODE AND CONFIGURATION

Several problems can arise when applications contain a mixture of data access code, business logic code, and presentation code. Such applications are difficult to maintain, because interdependencies between all of the components cause strong ripple effects whenever a change is made anywhere. High coupling makes classes difficult or impossible to reuse because they depend on so many other classes. Adding new data views often requires reimplementing or cutting and pasting business logic code, which then requires maintenance in multiple places. Data access code suffers from the same problem, being cut and pasted among business logic methods. The Model-View-Controller design pattern solves these problems by decoupling data access, business logic, and data presentation and user interaction.

Model-View-Controller(MVC) is a software architecture pattern which separates the representation of information from the user's interaction with it. In addition to dividing the application into three kinds of components, the MVC design defines the interactions between them.

Controller: a controller can send commands to its associated view to change the view's presentation of the model. It can also send commands to the model to update the model's state.

Model: a model notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. A passive implementation of MVC omits these notifications, because the application does not require them or the software platform does not support them.

View: a view requests from the model the information that it needs to generate an output representation.

SSH (struts, spring, hibernate, three frameworks for Java platform) is a classic MVC pattern. Actually Struts is a full MVC pattern. ActionServlet which is a part of Struts plays the controller role in MVC model. ActionForm and JavaBean play the model role. And JSP plays the view role. Spring framework is an open source application framework and inversion of control container for Java platform. It simplifies enterprise application development. Hibernate is an object-relational mapping library for Java platform, providing a framework for mapping an object-oriented domain model to a traditional relational database. Hibernate solves object-relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handing functions.

SSH as a classic MVC pattern, the three frameworks play different roles. Struts is responsible for the web layer. Spring is for service layer (or called manager layer). And Hibernate is for persistence layer. MVC-based SSH system in Figure 3.

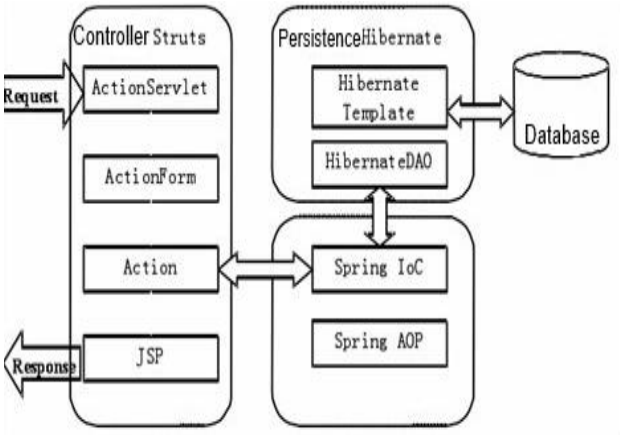

Figure 3. MVC-based SSH system

As shown in SSH framework Struts implements the MVC hierarchy, it makes JSP page, Action scheduling and specific business logic processing separated; Hibernate makes JDBC a very lightweight object package, which can manipulate the database arbitrarily using the object-oriented programming; Spring realizes interface-oriented programming using JavaBean, and provides many enterprise application functionality.

IV. A PROGRAM FOR IMPLEMENTING SOD

This project uses MySql as a background database. The main line is user - role - permission – resource, respectively corresponding to the many-to-many relationship. Three intermediate tables are used. Extended separation of duties requires a separate table to configure, User department also requires the department table associated with the user table. Login password for security purposes should establish a password table associated with user table.

Database table structure as shown below.

- User Table (users): number, user number, user name, gender, age, department number, description, the history of role
- The role table (roles): Number, role number, inherited role number, description, department number
- Permissions table (permission): Number, permissions number, description
- Resource Table (resource): Number, resources number, resource name, description
- corresponding table of Users - role (ua): Number, user number, role number, description, role time, role using total number, role using current number
- corresponding table of Role - permissions (pa): number, role number, permission number, description

- corresponding table of Permission - resource (pr): number, permission number, resource number, operation
- separation of duties enumeration table (sod): number, conflict roles 1, the conflict role 2, the separation of duties types, description
- administrator password table (customer): number, login name, password
- user password table (password): Number, user name, password
- departments tables (department): Number, department number, department name, description, department ext.
- Event table (event): number, event number, description, whether the event is closed, the event staff number, department number
- Event user table (eur): number, event number, event user number, the event user roles number. wether is activation
- Session table (sessions): number, session number, session user number, session time

User transaction processing system flowchart as shown in Figure 4.

The main function of foreground is to provide users with reasonable access resources programs, and to achieve the separation of duties rules. The foreground has event set, event selection, role selection and of resource operation. In which event set to judge whether the presence of the Minister of user roles permission, if not the authority of the Secretary can not set the event. Selection of the event, the first to determine whether the user Minister added to this event, if it has not added to the event, you can not enter this event. And then determine whether the user first log on to the event, if it is the first time you log in, you need to select the role in the event, in the role is selected, it is necessary to determine whether the role that has expired, and whether more than the frequency of use, if you choose a multi-a role, the need to determine whether the role of the role of the group have been included in the table dynamic separation of duties conflict.

The Admin main task is to modify users, roles, permissions, resources, separation of duties, as well as correspondence between most of the backend interfaces are used to configure these information. Information configuration page include: user password modification, department information changes, user information modify the role information modify permissions information changes, modifications of the resource information corresponding information to modify user roles, role permissions corresponding information to modify permissions to modify the resources corresponding information, static separation of duties allocated to modify, the historical separation of duties assigned to modify, dynamic separation of duties assigned to modify.

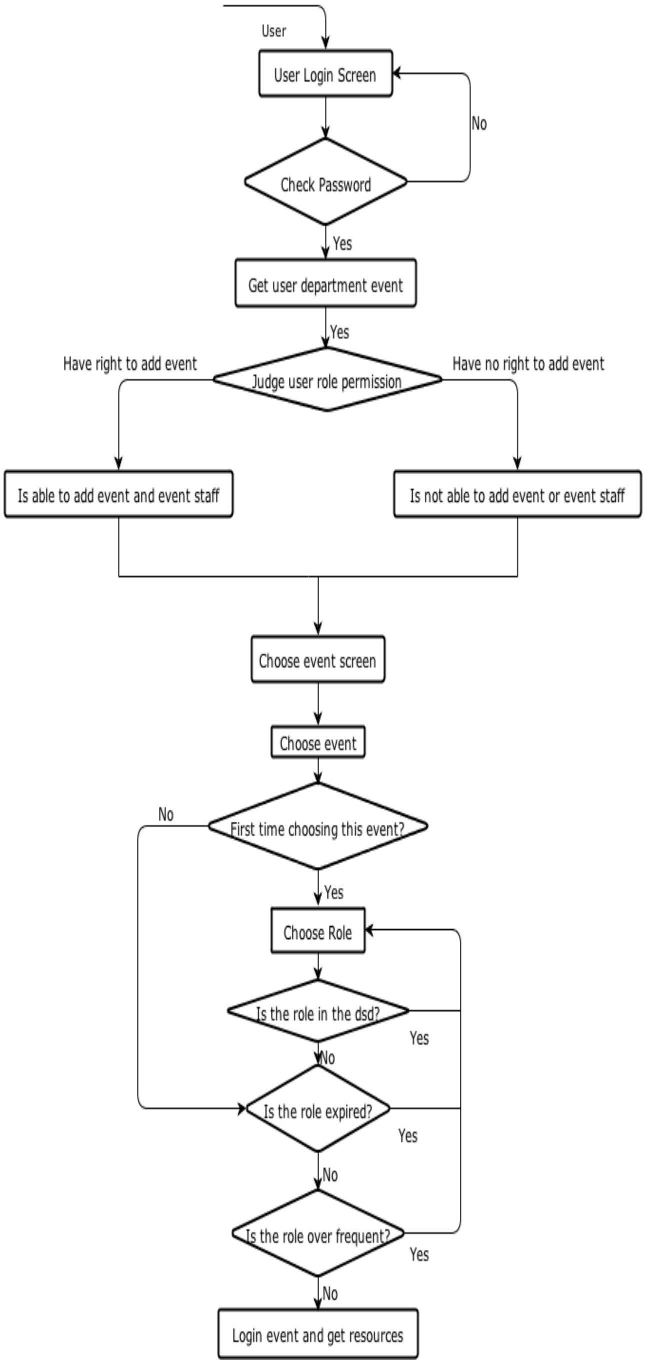
User
User Login Screen
No
Check Password
Yes
Get user department event
Yes
Have right to add event
Judge user role permission
Have no right to add event
Is able to add event and event staff
Is not able to add event or event staff
Choose event screen
Choose event
No
First time choosing this event?
Yes
Choose Role
Is the role in the dsd?
Yes
No
Is the role expired?
Yes
No
Is the role over frequent?
Yes
No
Login event and get resources

Figure 4. User transaction processing system flowchart

V. CONCLUSIONS

The main topic of the role-based access control mechanism is to solve the problem of the management of the information management system. This paper details the entire project from theoretical research to planning and design, and then to complete the final step. The foreground and background of the design is based on a three-tier architecture, mainly to solve the user how to obtain resources, and managers how to allocate resources. According to the needs of the MVC-based information management system security, this paper first researches separation of duties (SOD) in Role-Based Access Control, as well as its application in practice, and then gives a program to implement RBAC model as the framework of SSH. Finally, this paper describes how to realize the RBAC in a specific information management system. The implementation of the project is based on the Waigaoqiao Shipyard ERP, but because of the amount of data is too large, the interception of the part of the data used to test and validate the feasibility of the program. The paper has a certain theoretical and practical value.

VI. ACKNOWLEDGMENT

Here special thanks to the two foundations for funding the project. Research and Innovation Projects Of Shanghai Municipal Education Commission, project number is 11YZ212.

REFERENCES

[1] David Farraiolo and Richard Kuhn, "Role-Based Access Control," 15th NIST-NCSC National Computer Security Conference, 1992.

[2] Ravi Sandhu, "Lattice Based Access Control Models," IEEE Computer, 26:11, November 1993.

[3] David D. Clark and David R. Wilson, "A Comparison of Commercial and Military Computer Security Policies," Proceedings of the 1987 IEEE Symposium on Research in Security and Privacy (SP'87), May 1987

[4] Michael J. Nash and Keith R. Poland, "Some Conundrums Concerning Separation of Duty", IEEE Symposium on Research in Security and Privacy, May 1990.

[5] Jerome H. Saltzer and Michael D. Schroeder, "The Protection of Information in Computer Systems," Communications of the ACM, 17:7, 1975.

[6] Ravi Sandhu et al. "Role Based Access-Control Models", IEEE Computer, February 1996.

[7] Ravi Sandhu, "Rationale for the RBAC96 Family of Access Control Models." Proceedings of the first ACM Workshop on Role-based access control", February 1996.

[8] David Ferraiolo et al., "Proposed NIST Standard for Role-Based Access Control," ACM Transactions on Information and System Security, 4:3, August 2001.

[9] Simon, R. and Zurko, M. E. 1997. Separation of duty in role based access control environments [C]. In Proceedings of the 10th IEEE Workshop on Computer Security Foundations (Rockport, MA, June 10-12). IEEE Computer Society Press, Los Alamitos, CA,183–194

[10] Virgil D. Gligor, Serban I. Gavrila, and David Ferraiolo. On the formal definition of separation-of-duty policies and their composition [C]. In Proceedings of IEEE Symposium on Research in Security and Privacy, pages 172-183, Oakland, CA, May 1998.

[11] Gail-Joon Ahn & Ravi Sandhu, "Role-based authorization constraints specification," J. ACM Transactions on Information and System Security., 2000

[12] Chunyang Yuan et al., "A Verifiable Formal Specification for RBAC," Lecture Notes in Computer Science, 2006, Volume 4318/2006,

[13] Ferraiolo, D., Cugini, J., Kuhn, D. R. "Role-Based Access Control (RBAC): Features and Motivations" [C]. Proc. 1995 Computer Security Applications Conference, 241-248, December 1995.

[14] Ahn, G. -J. AND Sandhu, R. 1999. The RSL99 language for role-based separation of duty constraints [C]. In Proceedings of 4th ACM Workshop on Role-Based Access Control (RBAC '99, Fairfax, VA, Oct. 28-29). ACM, New York, NY, 43–54.

[15] G.J. Ahn, R. Sandhu. The RSL99 language for role-based separation of duty constraints. ACM Workshop on Role-Based Acces Control, Fairfax, Virginia, USA, 1999

[16] G.J. Ahn, R. Sandhu. Role-based authorization constraints specification. ACM Trans on Information and System Security, 2000, 3(4) :207-226

[17] D. R. Kuhn. Mutual exclusion of roles as a means of implementing separation of duty in role-based access control system. The 2nd ACM Workshop on Role-Based Access Control, Fairfax, VA, 1977

[18] N H Li, Q H Wang, M V Tripumitara. Beyond separation of duty: An algebra for specifying high-level security policies. Purdue University, CERIAS, Tech Rep: 2005-75, 2005

[19] Joon S.Park et al., "Role-based access control on the web s," J. ACM Transactions on Information and System Security, April 2001.

[20] Jean Bacon et al., "A model of OASIS role-based access control and its support for active security ,", J. ACM Transactions on Information and System Security., April 2002

[21] Jason Crampton, "Delegation in role-based access control ," J. International Journal of Information Security,2007.

[22] Zhang Zhiyong, "Collaboration Access Control Model for MAS Based on Role and Agent Cooperative Scenarios," J. IEEE International Conference on Mechatronics and Automation,2006