Implementing Web Access Control System for the Multiple Web Servers in the Same Domain Using RBAC Concept

Won Bo Shim[1], Seog Park[2]
Dept. of Computer Science, Sogang University
cool96@chch.ac.kr[1], spark@dblab.sogang.ac.kr[2]

Abstract

As the web server based system is being used more and more, having separate web servers for each task to distribute the web server's load are gaining much more popularity over having one main web server to process all the tasks. When the user tries to access each web server that contains number of web documents that are linked to each other via hyper-link within the domain, each web server asks the user to follow the verification process even though the user is identical, and this prohibits the user from using the system efficiently. Role based access control method, which is the most suitable access control concept available now for the distributed web server based system within the domain, will be used in this paper. Additionally, the method on controlling the level of web document contents available to the user based on the user's access permission rights will be introduced to reduce the granularity of the document content access.

1. Introduction

Access control in the distributed web system accompanies quite a few hardships due to the large number of the users and the web servers.

RBAC(Role Based Access Control) has been studied as the solution for managing resources in distributed environment, because it helps to reduce the number of errors occurred when managing the users and the network resources and also reduce the management fee[8].

In this paper we will introduce the method to realize the access control of the distributed web servers and controlling document view depending on the user's role by using the RBAC server information.
And by using memory cookie when the user accesses web servers in the same domain, the user can access transparently all the resources in multiple web servers without more authentication process at each web server as if those are all in one web server.

Ravi Sandhu and Joon S. Park from George Mason University published a paper titled "Secure Cookies on the Web" in 2000[4] which introduced the IP Cookie, Password Cookie, and Seal Cookie. But in this paper they focused on the design of cookie itself to guarantee the integrity of the cookie because they used the cookie which was saved in the hard disk.

2. Understanding the RBAC(Role Based Access Control)

2.1. Basic concept of the RBAC

The main idea of the RBAC is to prohibit the user from accessing the company's and/or organization's resources freely. Instead certain access rights will be assigned to the particular role, and the user can only access the minimum level of resources based on the user's role. The rights management can be simplified by this idea, and also it provides the flexibility when setting up the special security policy for the company. The user will belong to the particular role based on the user's task rights and responsibility, and the user can change the role without changing the access structure.

RBAC model concept supports the following well-known three security principles.[8][9]

Least privilege principle : Assigning the minimum permission only for the role needed to complete the task based on the role hierarchy.

Separation Of Duty : Frauds which could invade the information, and tasks which could cause unlawful means will be categorized as interactive supervisory role, and they are to be fulfilling their tasks separately.

Data Abstraction : Supports the commercial commands such as credit, debit, transfer, create account, and delete account which can fulfill variety of tasks and also can abstract the commands, instead of supporting

0-7695-1153-8/01 $10.00 © 2001 IEEE

the traditional commands such as 'read', 'write', and 'execute'.

2.2. Basic RBAC Model

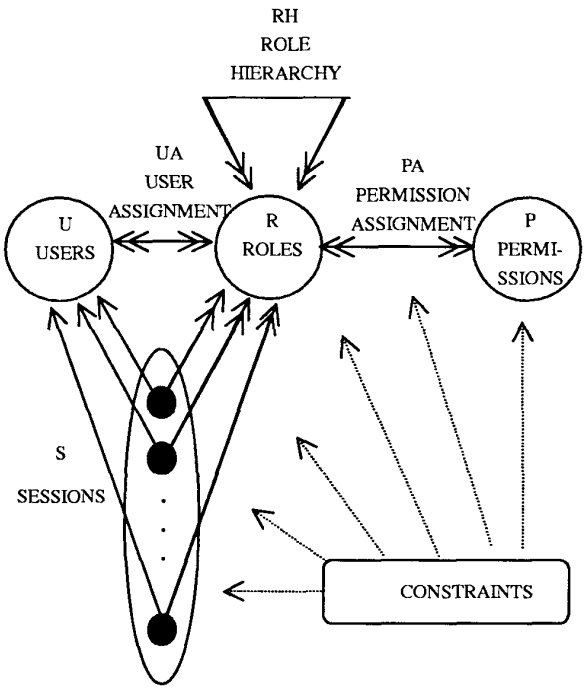


Figure 1. Basic RBAC model

Figure 1 shows the basic model of the RBAC. The basic model of the RBAC consists of users, roles, permissions, sessions, and constraints.[8][9][13]

User and Role: The user is the object that accesses the information in the computer system, and one user corresponds to one person. The role is very important meaningful structure in realizing the access control policy. System administrator creates the role and assigns the rights to the role based on the company's and/or the organization's tasks in the RBAC system.

Role hierarchy is defined as the partial order relationship between related roles, and because it is very similar to the right and the responsibility hierarchy system of the company, it can be used to model the right hierarchy of the company.

Permission : Permission is the approval of the special access mode such as read, write and update for one or more objects in system. Permission in the RBAC has the meaning of authorization, access right, and privilege.

Session : Session can be formed when the user sets the part of the role active by logging onto the system. One user is mapped to multiple rights in one session. Double arrow in Figure 1 indicates that the multiple roles became active simultaneously. The rights that are available to the users are all the rights combined set active by the roles in those sessions.

User assignment and Permission assignment : User assignment and permission assignment are referred to multiple vs. multiple relationship, and this is very important factor for RBAC model. One of the characteristic of the RBAC is assigning operations to the necessary roles for completing the task(permission assignment) instead of assigning operations to the direct users. The user becomes a member of the corresponding role(user assignment), and can complete the supported operations for the information object. This method provides the easiness of managing the rights in the company with many users and many information objects.

Constraints: Constraints can be used to reflect the policy of organization in RBAC operation. In order to prevent the wrongful deed, one user is not allowed to have exclusive roles at the same time(Static Separation of Duty) or one user can have exclusive roles but not allowed to perform both roles at the same time(Dynamic Separation of Duty), the number of users assigned to certain role is limited(Cardinality), and the prerequisite role for obtaining a role is defined(Prerequisite).

3. The roles of multiple web servers in the same domain.

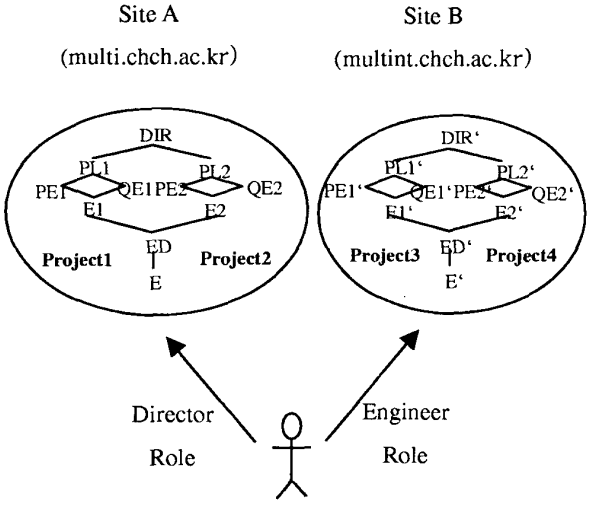


Figure 2. Different role at different site

Figure2 shows the RBAC concept in multiple web server environment. Site A can contain its own unique role structure, and site B can contain its own role structure which is different from site A. In this case, the same user will have different roles in each site.

For example, the user indicated in the Figure2, will have a director role in site A, and engineer role in site B. Then the user will obtain the access rights given to the director role in site A, and the access rights given to the engineer role in site B.

And Figure 3 shows that in site A the Director role user can see the entire document since the director role has the rights to browse the top secret level part in the document, but engineer role user can only see part of the document as an engineer since engineer role has the rights to browse only the confidential level part in spite of the same document.

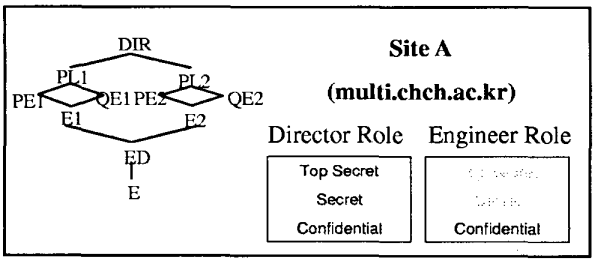


Figure 3. Different document view at the same site depending on the user's role permission

4. Accessing multiple web servers in the same domain

4.1. Scenario

For example, when the user requests the web document in site A, the web document including PHP script checks the validity of the user. A cookie can be used for this process.

The web document that the user attempted to access will be displayed on the browser if the user has the valid cookie which has been created after the successful verification process, and the verification process will be followed if the user has not been verified before. The user can receive a valid cookie from site A upon the successful completion of the verification process with the user's ID and the password.

The cookies are generally saved on the hard disk for the future usage, but this paper is written on the assumption that the cookies are not to be saved on the hard disk but only to be saved on the main memory so that the cookies are only valid while the browser is open in order to avoid the security issues. This also applies when the user visits site B first.

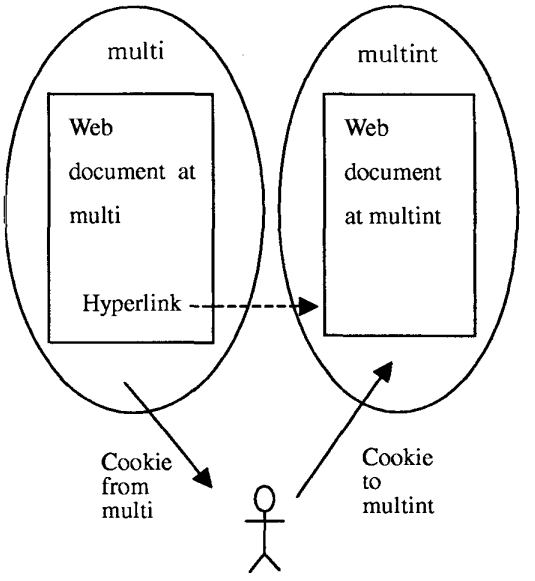


Figure 4. Hyperlink using RBAC cookie

Figure 4 shows that the additional verification process is avoided when the user accesses other servers in the same domain after the user receives the valid cookie from any site within the domain. The valid cookie in the main memory certifies the validity of the user, and web servers let the user accesses the web document with the appropriate control for their roles without additional verification processes.

4.2. Complete system diagram

Figure 5 shows the complete system diagram for this situation. Different multiple web servers can exist within the domain, and roles with their own role structures exist in each web server.

Role Server manages the role information and provides this information when each site asks it, and this could be distributed throughout the servers.

When the user tries to access a document in certain site, the user will be asked to follow the verification process by the corresponding web server.

A cookie is created in the main memory of the user's client PC as soon as the user is verified by the corresponding web server.

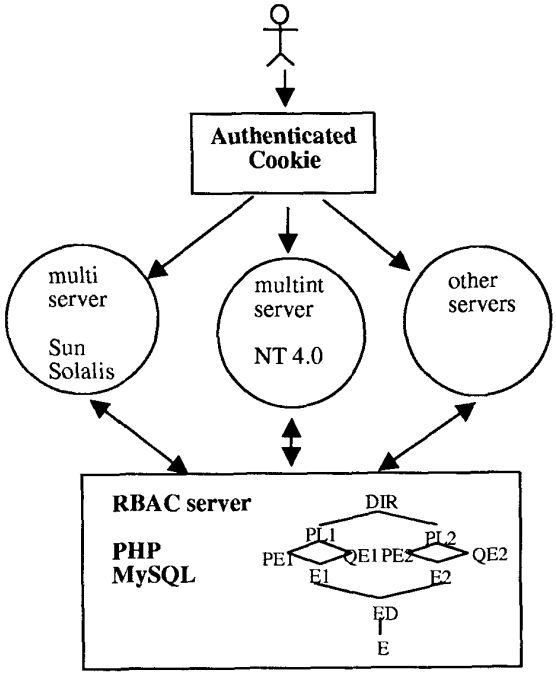
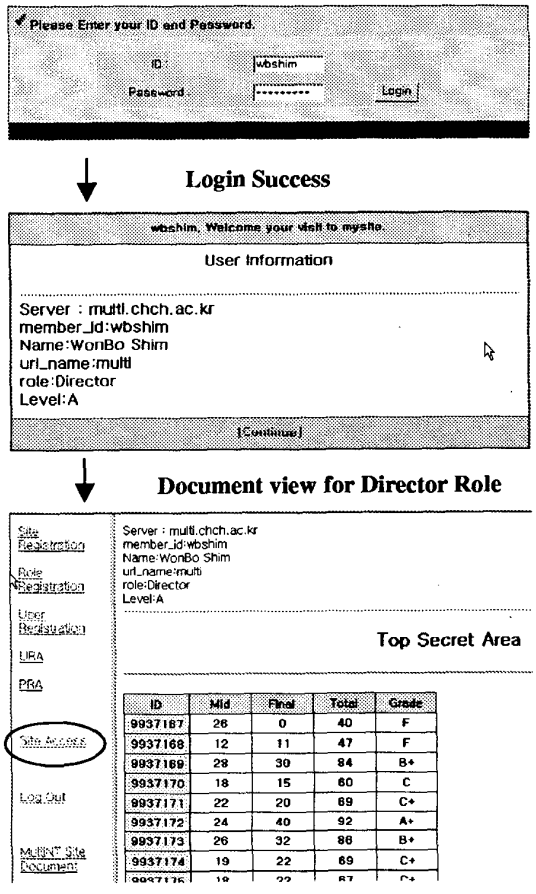


Figure 5. System Configuration

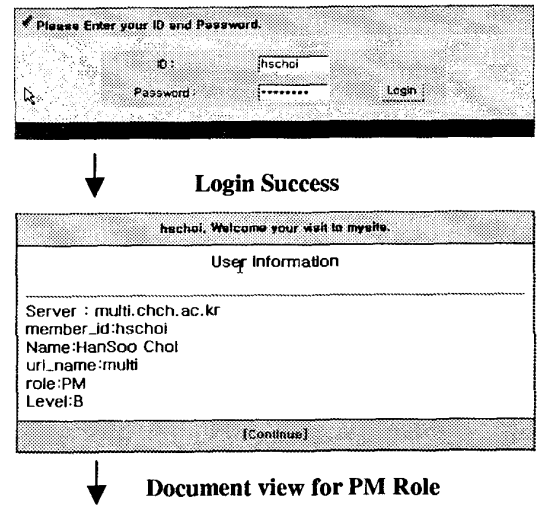
After the cookie has been created in the main memory of the user's client upon the completion of the verification process, the user can access other web servers without additional verification process by providing the valid cookie in its own main memory.

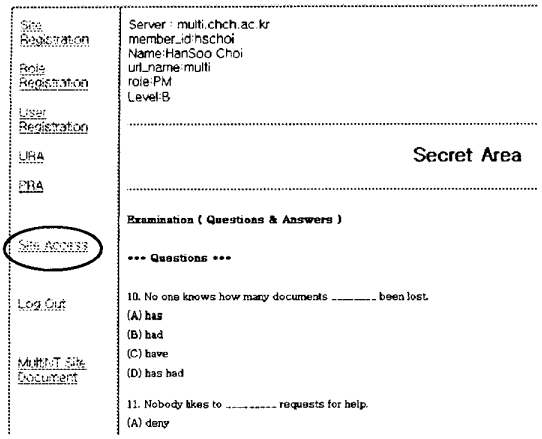
4.3. Implementing multiple web server access control in the same domain

In this part we want to show this paper introduces the reasonable, efficient and realizable access control method in multiple web server environment by implementing demo system. For this demo, I used PHP4 Server side script with HTML for the web documents and MySQL database system for RBAC DBMS.



a) Director Role User Login Process





b) PM Role User Login

Figure 6. Different view of the same web document according to the user role.

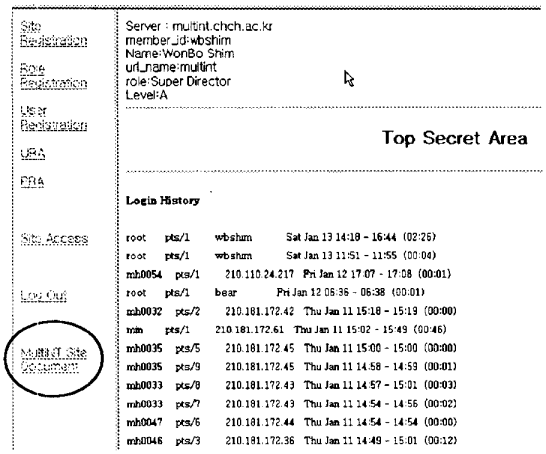


Figure 7. Access other site with the previous authenticated cookie

Figure 6 a) shows how to realize the multiple web server access control within the domain. When the user, wbshim, tries to access a web document in the multi web server, the user is asked to verify himself, and upon the successful completion of the verification, the role server transfers the information of the user's role and access rights to the user in cookie format. Then the user can see his role information showed in second screen in figure 6 a), and because the current user's access right is the highest role, Director, the user can view the entire part including top-secret area in the web document.

In Figure 6 b), because hschoi's access right is lower than this, PM which was defined in RBAC server, the user can view secret part only that is one step lower.

Figure 7 shows how the user can access the other multint server (multint.chch.ac.kr) via hyper link within the domain after successful logging onto the initial multi web server (multi.chch.ac.kr). The multint server checks whether the user has been verified before based on the cookie, and because the user has received the valid cookie already when accessing the initial multi server, the user can see the documents in multint server without another verification process.

In this situation, Role server provides the appropriate access rights for the user, because the user's role varies in the multint server and therefore the user's access rights varies along.

Additionally, the role server provides the site information, the role information of the site, the user information, permission role assignment, and user role assignment to each web server.

With this information and the suggested scheme we can achieve the access control in multiple web server environment efficiently and conveniently for users.

5. Conclusion

Efficient access control is needed in the distributed web system due to the large number of the users and the web servers. The RBAC concept has been used to reduce the number of errors occurred in resource management between the users and the network resources and also reduce the management fee.

We suggested how the large number of users can access the distributed web servers efficiently. The users can access multiple web servers transparently within the domain without being asked the redundant verification after successful login to the initial web server, and how the task could be completed more efficiently using this method which was shown in this simple demo system.

In addition to that, we suggested the idea to provide the different view to the same web document

depending on the access rights based on the user's role permission.

Acknowledgements

This work was supported by grant No.2000-1-303-001-3 from Basic Research Program of the Korea Science and Engineering Foundation.

References

[1] D. Ferraio, J. Cugini and R. Kuhn, "Role-based Access Control(RBAC): Features and motivations", Proc. of 11th Annual Computer Security Application Conference, (1995).

[2] John Barkley, "Managing Role/Permission Relationships Using Object Access Types", Third ACM Workshop on Role-Based Access Control., (1998).

[3] R. Sandhu and Gail-Joon Ahn, "Group Hierarchies with Decentralized User Assignment in Windows NT", ASTED-CSE, (1998).

[4] Ravi Sandhu and Joon S. Park,"Secure Cookies on the Web", IEEE Internet Computing, July-August, (2000)

[5] R. Sandhu and V. Bhamidipati, "The URA97 Model for Role-Based User-Role assignment", Proc. of IFIP WG 11.3, (1997).

[6] R. Sandhu and Q. Munawer, "The RRA97 Model for Role-Based Administration of Role Hierarchies", ACSAC, (1998).

[7] J. F. Barkely, A. V. Cincotta, D. F. Ferraiolo, S. Gavrilla and D. R. Kuhn, "Role Based Access Control For the World Wide Web", 20th NCSC, (1997).

[8] R. Sandhu, E. Coyne, H. Feinstein, and C. Younman, "Role-Based Access Control Models", IEEE Computer Magazine Vol. 29, (1996).

[9] Sejong Oh, Seog Park : Task-Role Based Access Control(T-RBAC):An Improved Access Control Model for Enterprise Environment, DEXA (2000)

[10] Ravi Sandhu, "Lattice-Based Access Control Models", IEEE Computer Vol 26, (1993)

[11] M. Nyanchama and S. Osborn, "The Role Graph Model and Conflict of Interest", ACM Transactions on Information and System Security, vol.2 (1999)

[12] Gail-Joon Ahn, R. Sandhu, Myong Kang and Joon Park, "Injecting RBAC to Secure a Web-based Workflow System", Proc. of the Fifth ACM Wrokshop on Role-Based Access Control, ACM, (2000).

[13] Ravi Sandhu, David Ferraiolo and Richard Kuhn, "The NIST Model for Role-Based Access Control: Toward A Unified Standard", Proc. of the Fifth ACM Wrokshop on Role-Based Access Control, ACM, (2000).

[14] E. C. Lupu and M.S. Sloman, "Reconciling Role Based management and Role Based Access Control", Second RBAC Workshop, ACM, (1997)