International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN : 2278-800X, www.ijerd.com
Volume 5, Issue 5 (December 2012), PP. 55-59

Scalable Transaction Authorization Using Role Based Access Control for Time Based Content Access with Session Management

J.Saravanesh[1], Dr.E.Ramaraj[2],
[1] Assistant Professor, Department Of Computer Science, MKU College, Madurai, Tamil Nadu, India
[2]Professor, Department of Computer Science, Alagappa University, Karaikudi Tamil Nadu, India

Abstract:- Information Accessing is vital part of Information Processing. Information Access is a boundless service in this digital era. Information accessing has three major considerations-How the Information is accesses, Who Accessed the Information and how long it needs to be accessed. It also relies on the type of information needed at the instance and the type of users who may be a single access or a group. The primary RBAC access control model which is widely used, does not efficiently address the time management for scalable transactions. This Paper proposes a RBAC strategy for providing time based user to the content. Such Time based Session management would provide Valid Privileges for the user and the content.

Keywords:- RBAC, Scalable transaction, Content access, Time base Content access, Information retrieval

I. INTRODUCTION

The Impact of Role base information access plays a vital role in information sharing. The need for RBAC further increases as there was a phenomenal growth in Information Technology via Internet. The Concept of RBAC [1] proposed by Ferraiolo and Kuhn in order to find solution for scalable issues arise in terms of Content access. This RBAC essentially provides roles for each and every user and the accessing were done as per the role devised. An Application were defines as a part of role for accessing the content. This application role provides valid user access roles. The permission for accessing the information's was done as per the role devised by RBAC. The RBAC were devising the role to content for accessibility and as it will not define the time constraint for access these information's.

The steps involved in assigning user role as per Seth Freeman [2] were as follows [i] User Role Assignment (URA) [ii] Permission role assignment (PRA). These two methods were focused to devise the role access based on the user and their roles. As per this standard the Role has to be set before the permission were provided for accessing the content.

The architecture of RBAC was focused on User Pull Architecture [3] where this architecture mainly addresses on pulling the content information from the server. Every architecture for RBAC consists of numerous clients, server, and in addition to that role server. Each and every component in the RBAC does their job of submitting the request from client, processing and storing this information in server and accessing the information from the server using role server. This is the predefined architecture for every RBAC.

The kind of architecture which was discussed in [5] deals with different version of RBAC based on the support extends towards User Role Assignment and as well Permission Role Assignment. The interface support by the different version of RBAC works on the principles of Object Oriented access. The need of Object Oriented paradigm focuses on the user role and invoking permission.

The following diagram will explain the user role access which explains UPA as discussed in [3].

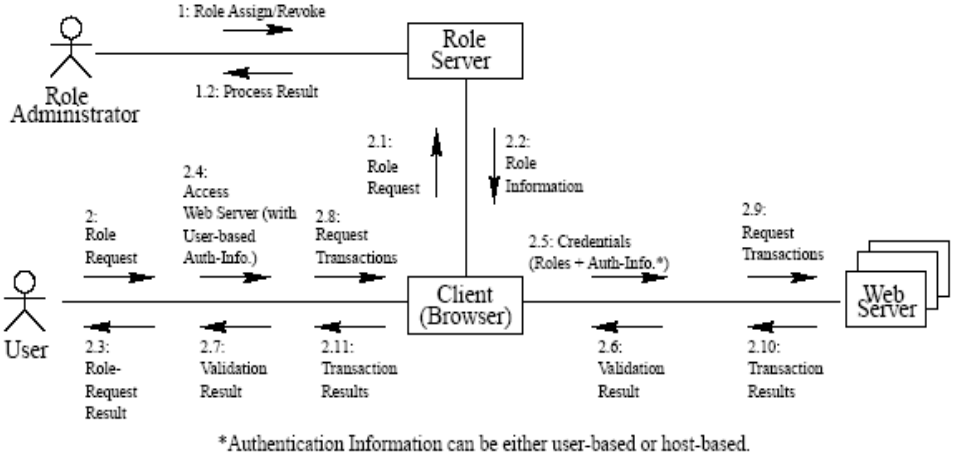

Fig.1 User Roll Access in User Pull Architecture

This figure discusses the work flow of user roll access along with the user categorization. This model distinguish user as user and admin where both have different role to play. Different roles in the sense, each role have its own way of accessing the information. The role of administrator will define the role on the role server and user will get the grant from the role server. Once the role is defined then the user may access the content from the web server.

Another way for providing user access comes from the server pull architecture.

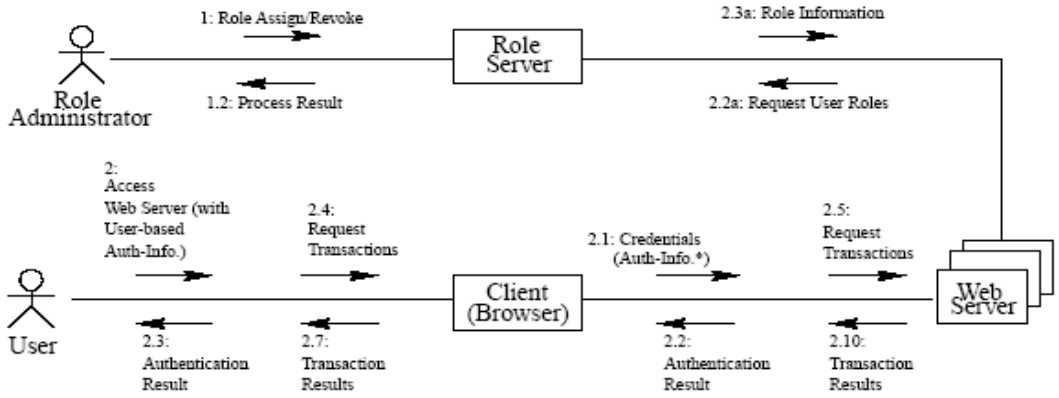

Fig.2 User Roll Access in Server Pull Architecture

In this figure the user get the access from role server and gets the service access grant web server. The role server will get the acknowledgement from the web server before providing access to the user. Both the User pull and Server pull architecture works based on the content and role. The time parameter aspect of user and server pull where not proven in detail.

The need for time sync plays a vital role in both server and user pull architecture. When both the architecture focus on roles and content and time will be an added parameter for ensuring the time taken or time spend for sharing or accessing information in both the architecture.

II. REVIEWED ARTICLES

The work of Joon S. Park, Ravi Sandhu, and Gail-Joon Ahn [6] focused on RBAC on Web and they also made a comparison of User and Server Pull Architecture. Another work of Joon S. Park, Ravi Sandhu, and SreeLatha Ghanta [7] focused on RBAC on the Web by Secure Cookies. The need of security based on cookies information will gather more secure information from the server to the user. This work also focuses on Cookie information by which user access will pertain in the server. The work of D.Ferraiolo and R. Kuhn [1] on Role-Based Access Control provides the basic need for invoking RBAC in Web. The work of Joon S. Park and Ravi Sandhu [11] address the need of smart certificate which acts as a secure certificate on web. Further in their work they have addressed on secure cookies information for the data availability on the net. The impact of LDAP on

RBAC on the web using LDAP by Joon S. Park and Ravi sandhu proposes a new strategy in the form of LDAP which focus on directory access protocol for RBAC.

III. EXISTING WORK

The work of RBAC plays a vital role in web accessing and also provides security based access for the data and to the system. The systems discussed in RBAC were as follows Client, Server, and Role server. The RBAC Model discussed by Joon S. Park and Ravi Sandhu [11] was represented by the Figure 3.

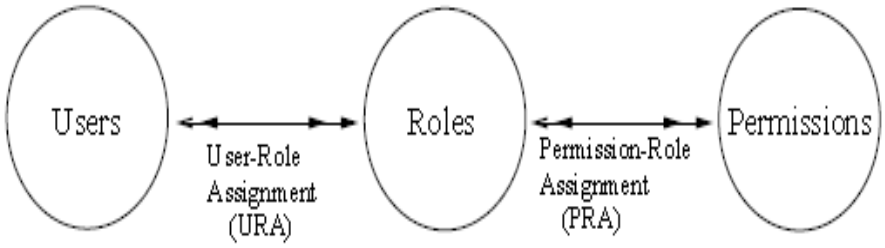

Fig 3: Simplified RBAC model

The RBAC model discussed above, focus on the roles and its permission. This model will address the types of user, their roles and permissions. The types of users are user, administration. User will have normal permission and based on the permission the role is defined where as the admin role will have complete access and he supposed to be the person to define access to each and every user. This model will never address the scalable portions that arise in the aspect of time compatibility. Though it provides the URA and PRA it will never reveal the time complexity principles arise during the momentum. Further this concept will never address the possibilities of user mismatch and authenticate user. This concept discussed above were the work on LDAP towards RBAC. Different user had different role to be played for accessing the LDAP. The problem is how an unauthenticated user was stopped from accessing the system. The time parameter discussed above will stop an intruder without depending on secure key concept and cookies as discussed in other work of RBAC i.e how an unauthenticated user kept in the system for making the login attempt successful. The problem faced in this method will address by our work.

IV. PROPOSED WORK

We have proposed a RBAC method was the time parameters were also be considered as a part of the system. This time constraint can very well adapted for both User Pull and Server pull architecture.
To explain further, our concept was explained with a Figure4 shown below.

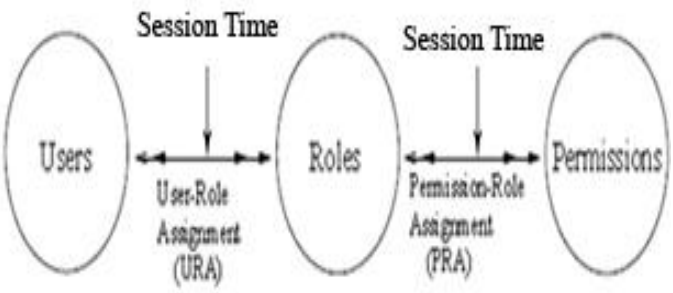

Fig 4: Session timing in RBAC model

In the above diagram the model was altered based on session time. The impact of session time gives the user some specific time by which a valid user can able to access the system. The procedure for deploying the session time strategy in URA and PRA were discussed below.

Procedure for session time in URA
Step 1: Initiate the login process
Step 2: Initiate the session time to access the number of attempts made by the user
Step 3: Once successful URA will define the role
Step 4: If unsuccessful Role will not be defined and user will be force fully logout.
Step 5: End result only successful sign in leads to role definition.

Procedure for session time in PRA

Step 6: Successful login leads to PRA
Step 7: Session value have to be retrieved for user comparison
Step 8: If the session value not matched with the user no permission will be allotted
Step 9: If matched permission will be defined.
Hence Session time plays a vital role in defining URA and PRA.
This strategy will be applied for User Pull and Server Pull Architecture with session timing.

Simulation result

	User Pull Architecture	**Server Pull Architecture**
User Convenience	Low	High
Performance	High	Low
Reusability	High	Low
Role Freshness	Low	High
Single point failure	Low	High
Session time failure	Low	Low
Session time success	**High**	**High**

Table 1: Comparison of User Pull and Server Pull Architecture

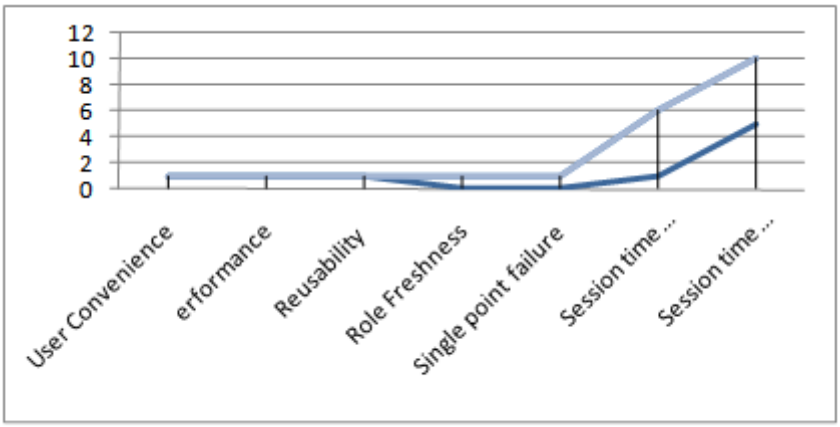

Figure 5: Impact line curve of Session time success and failure in RBAC model

The graph shows that the impact of session time success is stay on positive note and it proven to be the wise way for adapting it in RBAC model for providing valid access to the user based on timing. We have made 5 sec as the time parameter to validate the user after keying the information and for every 5 seconds the count happens.

V. CONCLUSION AND FUTURE WORK

In this work we have addressed the need the session time for ensuring valid user is getting accessed to the site. The impacts of security key were not considered as a parameter since we have fixed the time scalable as a measure to access the site. The future works of ours will address the need of security key along with session time since the data has to be passed in the network.

REFERENCES

[1]. [1] D. Ferraiolo and R. Kuhn. "Role-Based Access Control," in Proceedings of 15th National Computer Security Conference, October 1992.
[2]. [2] Seth Freeman. "Role-Based Access Control on the Internet" Summary of Two Selected Articles on RBAC, May 2005.
[3]. [3] J. Park, R. Sandhu, and G. Ahn. "Role-based Access Control on the Web," ACM transactions on Information and System Security, 4(1), February 2001.

[4]. [4] D. Shin, G. Ahn, and J.Park. "An Application of Directory Service Markup Language (DSML) for Role-based Access Control (RBAC)," in Proceedings of the 26th Annual International Computer Software and Applications Conference, August 2002.

[5]. [5] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. "Role-Based Access Control Models," IEEE Computer, 29(2), February 1996.

[6]. [6] Joon S. Park, Ravi Sandhu, and Gail-Joon Ahn. RBAC on the Web. ACM Transactions on Information and Systems Security, 4(1), February 2001.

[7]. [7] Joon S. Park, Ravi Sandhu, and SreeLatha Ghanta. RBAC on the Web by Secure Cookies. In Proceedings of 13th Annual IFIP11.3 Conference on Database Security, Seattle, Washington, July 1999.

[8]. [8] Nicholas Yialelis, Emil Lupu, and Morris Sloman. Role-based Security for Distributed Ob ject Systems. In Proceedings of IEEE Fifth Workshops on Enabling Technology: Infrastructure for Col laborative Enterprise. 1996.

[9]. [9] Netscape Communications Corporation. Netscape Directory Server 4.1 Deployment Guide. http://developer.netscape.com/docs/manuals/directory/dir40/de/contents.htm, 1999.

[10]. [10] Joon S. Park and Ravi Sandhu. Secure Cookies on the Web. IEEE Internet Computing, 4(4), 36-44, July-August 2000.

[11]. [11] Joon S. Park and Ravi Sandhu. Smart Certicates: Extending X.509 for Secure Attribute Services on the Web. In Proceedings of 22nd National Information Systems Security Conference (NISSC), Crystal City, Virginia, October 1999.