ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

A survey paper on Role Based Access Control

Dipmala Salunke, Anilkumar Upadhyay[1], Amol Sarwade[2], Vaibhav Marde[3], Sachin Kandekar [4]

Student,Information Technology,RSCOE,Pune,India[1]

Student,Information Technology,RSCOE,Pune,India[2]

Student,Information Technology,RSCOE,Pune,India[3]

Student,Information Technology,RSCOE,Pune,India[4]

Abstract: This paper presents the Role Based Access Control framework which deals with safe and security access to the system resources. The RBAC framework is used to specify the rules about how to set up the process for granting or denying authorizations. The basic cornerstones of RBAC are authentication, authorization, assigning permissions and session management. The existing systems did not contain access for non Active Directory users which is the main emphasis of our project along with providing dynamic alteration of user abilities. This project includes the use of Permission objects in order to tackle with permission level security and also we are using the Data Access Layer concept to make it specific database independent.

Keywords: RBAC, authentication, authorization, permissions, roles, session, users

I. INTRODUCTION

The prominent progress of Internet and related technologies has promoted tremendous information and data sharing. There is a growing concern for security and privacy of data, and numerous studies have shown that unauthorized access can cause great losses, especially in financial software. Access control becomes more and more essential for safe and secure access to the software and hardware resources. RBAC model provides a powerful way to satisfy the access control needs. An access control policy is a statement which specifies the rules about how to setup the process for granting or denying authorizations to the users.The concept of *role* is central to RBAC. As defined by the standard, "a role is a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role. RBAC's fundamental rationale is that a role is an intermediate element between users and permissions. An RBAC implementation directly assigns users to roles (many-to-many assignments) and permissions to roles (many-to-many assignments), and thus indirectly assigns users to permissions. According to the standard, "the permissions available to the user are the permissions assigned to the roles that are currently active across all the user's sessions." A *user* is normally considered to be a human being, but it could also be a process, machine, or network. A *permission* is an approval to perform an *operation* on an object—that is, an action, function, or task that a user can invoke. The term *object* can refer either to information containers (such as files, directories, or database tables) or resources (such as printers, network drivers, or computers). A *session* is a mapping between a user and a set of assigned roles, with one-to-many user-session assignments and many-to-many session-role assignments. RBAC allows a user to activate multiple roles simultaneously in a single session, although it is not necessary to activate all roles. Sessions implement the core RBAC model, which supports many-to-many user-permission assignments.

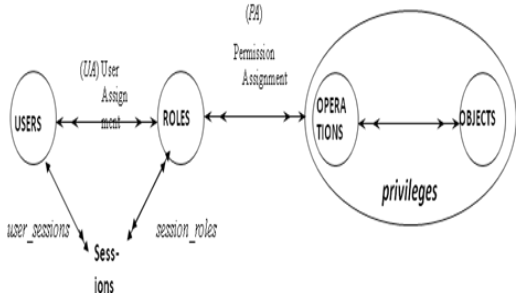

Fig 1 .working of RBAC

Also part of the core RBAC model are two functions to review the set of users assigned to a given role and the set of roles assigned to a given user. Other review functions in the standard are advanced, implying they are not mandatory.

II. LITERATURE SURVEY

There are a number of existing systems based on the concept of roles and permissions in the market and we will take a closer look at those systems.

A. DAC

DAC is a kind of access control model which can allow subjects to grants certain restriction to access. It's based on access matrix model[1]. DAC allows subjects grant or revoke access privilege to the objects which belong to them. This makes access control discretionary, which is the main disadvantage of DAC. Management of access control is very discrete. Relationship among clients in system cannot be displayed clearly, which also makes management very difficult.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

Discretionary Access Control (DAC) allows authorized users to change the access control attributes of objects, thereby specifying whether other users have access to the object. A simple form of Discretionary Access Control (DAC) might be file passwords, where access to a file requires the knowledge of a password created by the file owner. In Linux, the file permission is the general form of Discretionary Access Control (DAC). Discretionary Access Control (DAC) is the setting of permissions on files, folders, and shared resources. The owner of the object (normally the user who created the object) in most operating system (OS) environments applies discretionary access controls. This ownership may be transferred or controlled by root/administrator accounts[1]. Discretionary Access Control (DAC) is controlled by the owner or root/administrator of the Operating System, rather than being hard coded into the system. The Discretionary Access Control (DAC) mechanisms have a basic weakness, and that is they fail to recognize a fundamental difference between human users and computer programs.

B. MAC

MAC is also called Latices-based Access Control, which is designed for stricter and more secure access control model than DAC[2]. In MAC model, system assigns a special security attribute to subject and object. Generally, a subject can't change the security attributes of another subject. It is the system that decides whether the subject has right to access object by comparing the security attributes of subject and object[2]. The traditional discretionary and mandatory access controls (DAC and MAC**,** respectively) are inappropriate for the information security needs of many organizations[1][2]. RBAC has been proposed **as** an alternative, and supplement, to traditional DAC and MAC. MAC, however, is not without serious limitations. The assignment and enforcement of security levels by the system under the MAC model places restrictions on user actions that, while adhering to security policies, prevents dynamic alteration of the underlying policies, and requires large parts of the operating system and associated utilities to be "trusted" and placed outside of the access control framework.

III. RELATED WORK

Today web services have been mostly used by different industries to improve the productivity and automate their customer support. So the RBAC is used to provide flexible security and access control to organizational information and resources. This can be implemented by providing authentication, authority and audit control. The basic idea of RBAC is grants access privilege to a certain role. User will play some roles in a system in order to get privileges. RBAC divides roles by its responsibility and duty which are relatively stable in a system. The systems only grant roles to users. The roles become bridge between subjects and objects in access control mechanism. The major concepts in this system are the use of a hierarchy used for the assignment of roles and permissions along with making the framework independent of any specific database so that it can be used on any of the databases. The additional support to active and non-active users is also proposed in the system and a mechanism to assigning permissions to a group of users depending on their roles.

A. Authentication

This confirms the user's identity. It checks the user identity by the application. This is a two step method in this first who are you? is checked by using the related information of the user and then the authentication will be provided i.e. who are you; this can be implemented by providing the username and password. It can be treated as a low level security.

B. Authorization:-

Authorization means what a user can do. In this step the actual security to the information and resources of the organization is provided by assigning the roles to the user but sometimes it is very good that providing an access to a user dynamically that means at the time of authentication the roles are not assigned to the user. The roles are provided when the request is send by the user to the admin. Then the admin checks all the permission and policies to the requested method if the user has access on to that particular requested method then and then only the request is granted else the message will be shown i.e. the access is limited. This step is heaviest than the authentication because all the access control and permissions are implemented by coding.

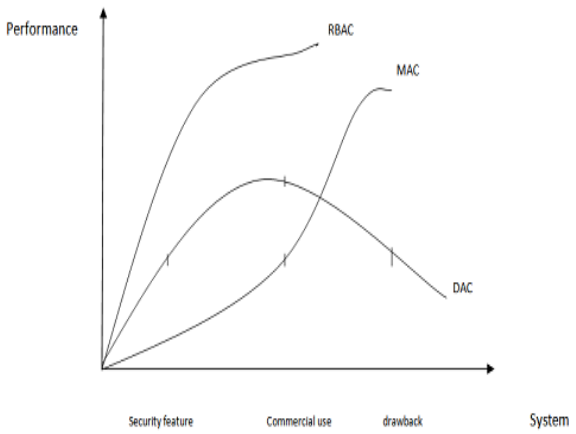

Fig 2 .comparison of DAC,MAC and RBAC

C. Audit Control:-

Keep tracking of the sensitive transaction is known as an audit control. Audit should enable you to review who did

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

what in your application, when and who granted which permissions to which user.

D. Persistence Schema:

The schema should have to be persisted after each transaction and it should not have to be change in between transactions.

E AD/Non-AD Users:

AD user means active directory user. Those user having or working in the same organization. These users are having a access to the internal information of the organization as per their role assigned in an organization. But also here in RBAC we are going to provide the access to the non AD user on the information of the organization as per their requirement. Again before providing the access to the non AD user all the permissions and condition are going to be checked. non AD user all the permissions and condition are going to be checked.

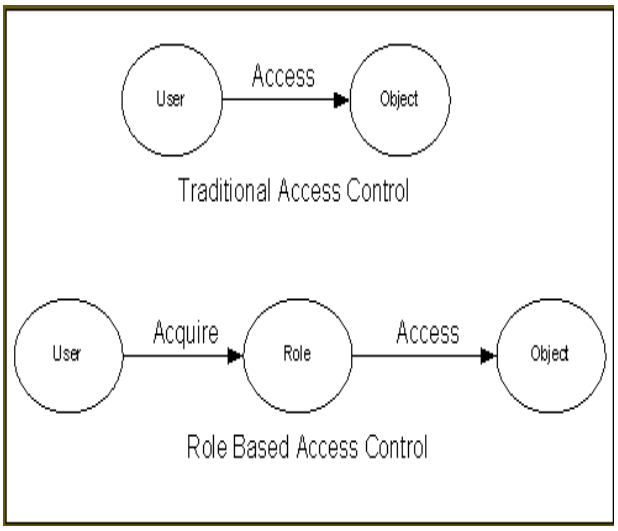

Fig 3 .framework of RBAC

V. CONCLUSION

Access control decisions are often based on the roles individual users take on as part of an organization. A role specifies a set of transactions that a user or set of users can perform within the context of an organization. RBAC can be quite effectively used by a number of organisations in order to maintain the security and stability of the organisations. Undoubtedly the techniques discussed above are extremely useful, a next step in this path would be to compare and evaluate all these various mechanisms by creating sets of data and an experimental testbed or to come up with a collaborative approach to find more efficient solution for RBAC framework.

VI. REFERENCE

[1] R. Anderson. Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley Computer Publishing, New York, New York, 2001.
[2] D. Bell and L. LaPadula. Secure computer system: Unified exposition and multics interpretation. TR M74-244, March 1976.
[3] Yang Jijiang, He Wei. Strengthen Top-level Design to Promote In-depth Development of E-government. E-Government, 2006, (12):150-157
[4] Zeng Zhongping, Li Zonghua, Lu Xinhai. The Access Control Policy Study of E-government Information Resource Based on RBAC. Journal of Information, 2007, (10):39-41
[5] Zhang Xiaoyan, Zhang Suwei. Design and Realization in E-government System Based on RBAC Theory.Computer Engineering and Design.
[5] Zhang Xiaoyan, Zhang Suwei. Design and Realization in E-government System Based on RBAC Theory.Computer 2007, 28(3):680-682
[6] Ferraiolo D, Kuhn R. Role-Based Access Control. Proceedings of the NIST-NSA National Computer Security Conference, 1992:554-563
[7] Sandhu R, Coyne E. Role-based access controlmodels.IEEE Computer,1996, 29(2): 38-47 [8] Barka E, Sandhu R S. Framework for role-based delegation models. In: Proc. of the 16th Annual Computer Security Application Conf. IEEE Computer Society Press, 2000: 168-176
[9] Xu Hongxue, Liu Yongxian.Temporary and domain role-based delegation model. Computer Applications, 2006, 26(2):323-326