Extensions to the Role Based Access Control Model for Newer Computing Paradigms

Ramadan Abdunabi
Colorado State University
Computer Science Department
rabdunab@cs.colostate.edu

October 26, 2010

Abstract

Authorization and access control are two of the most important techniques to ensure the security of systems. There has been a lot of work done by security researchers in developing flexible and semantically rich authorization / access control models. Currently, the Role Based Access Control Model is the de facto standard. However, notwithstanding its popularity, RBAC has been found lacking in many computing applications. Consequently, resarchers have proposed numerous extensions to the classical RBAC model. Unfortunately, we find in this work that there are quite a few newer types of applications that impose five authorization requirements at the same time, that are not satisfied by any of the proposed extensions of RBAC. We outline a new authorization model to fill this gap and conclude that there is still need of continued research in this area.

Keywords: Authorization, RBAC, Trust, Delegation, Context

1 Introduction

Authorization and access control has always been a fundamental security technique in systems in which multiple users share access to common resources. Authorization is the process of expressing security policies that determine whether a subject (e.g., process, computer, human user, etc.) is allowed to perform an operation (e.g., read, write, execute, delete, search, etc.) on an object (e.g., a tuple in a database, a table, a file, a service, and, more generally, any resource of the system). These policies define the subject's permissions (rights to carry out an operation on an object) in a computer system. Access control is the process of enforcing these policies in order to achieve the desired level of security. Although, access control and authorization are distinct techniques, the terms are often used interchangeably. We will do the same in this paper.

Managing and administering the users' privileges is one of the most challenging task in access control. Several access control models have been proposed, such as, discretionary and mandatory access control models (DAC and MAC), Clark-Wilson model, Lipner's Integrity model, Chinese Wall model, Task based models, and Role Based Access Control models. Among these models Role-based access control (RBAC) models have been receiving attention as they provide systematic access control security through a proven and increasingly predominant technology for commercial organizations. One of the main advantages of the RBAC over other access control models is the ease of its security administrations [4]. Using RBAC, organizations are capable of modeling security from their unique perspective. RBAC models are policy neutral [5]; they can support different authorization policies including mandatory and discretionary through the appropriate role configuration.

In spite of the success of the RBAC, researchers have determined that there are still many application security requirements that are not addressed by the classical RBAC models [6]. In the past few years, several RBAC extensions have been proposed to address such security requirements [2, 3, 4, 6, 11, 16, 17]. Although, these extensions are geared towards specific application requirements, we find some applications where individually these extended models fall short. These applications require the combined capabilities of some or all of these extensions.

In this paper, we begin by identifying an application where traditional RBAC fails. We investigate the authorization requirements imposed by this application. We then survey some of the newer extensions to RBAC that can potentially address these requirements individually. We posit that there is a requirement for an intergrated extended RBAC model to address such a new application. We then identify some of the challenges involved in incorporating the various extended RBAC models into an integrated whole.

The rest of the paper is organized as follows. Section 2 provides a background of the classical RBAC model. Section 3 describes a new application that imposes five novel authorization challenges. Sections 4.2 – 4.5 outlines some of the more important RBAC extensions. Section 5 outlines the challenges that remain in incorporating

different RBAC extensions into an integrated whole. Finally, section 6 concludes the paper with discussions about future work.

2 Preliminaries of Role Based Access Control Model RBAC

Sandhu et al [1] proposed RBAC96 which is a family of four constitutes models. In RBAC permissions are associated with roles (the intermediate concept of roles can be seen as collections of permissions), and users are made members of appropriate roles. The notion of role is an enterprise or organizational concept. The definition of role is quoted from Sandhu et al. [1]: A role is a job function or job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role. Permissions are not directly assigned to users, instead they are assigned to roles. RBAC comprise a family of four references models:

RBAC0: contains the core concepts of the Model. It is the minimum requirement for any system that exploits features of RBAC. Users (U), roles (R), and permissions (P) are three sets of entities and the relations between these entities are defined by User-Role Assignment and Permission-Role Assignment [1]. These sets and relations are the main concepts of the RBAC. A user can be member of many roles and each role can have many users. A user can invoke multiple sessions within a session a user can invoke set of roles but each session belongs to only one user. Permission can be assigned to many roles and a role can have many permissions.

RBAC1: adds to RBAC0 a role hierarchy (RH). Role hierarchies are an important concept for structuring roles to represent organization users responsibly and degree of authority.

RBAC2: introduces the concept of constraints. RBAC adds static (not related to sessions) and dynamic (related to sessions) constraints between core concepts [1] . These constraints are considered to be the principle motivation for RBAC because constraints are powerful mechanism to lay out higher-level organizational mechanism [1]. Constraints can be applied to User-Role Assignment, Permission-Role Assignment and session.

RBAC3: includes all aspects of RBAC0, RBAC1 and RBAC2 and it is called a unified model of RBAC. RBAC3 combine RBAC1 and RBAC2 to combine both role hierarchy and constraints. In this model constraints can be applied to the role hierarchy in addition to the constraints in RBAC2.

3 Authorization Challenges in Newer Computing Paradigm

Newer computing paradigms such as pervasive computing, mobile computing, and cloud computing imposes novel challenges for authorization that are not addressed by traditional RBAC.

Consider the following real world example from the healthcare domain. Suppose a healthcare system of a medical-care facility is used by physicians, nurses, patients, and administrators. This system needs the following security requirements: (1) Physicians from other medical-care facilities can access to the system and obtain medical information about their patients. The identities of such physicians are not known in advance. Further, patients' medical information are released to the physicians based on their historical information. (2) Nurses play a role of day-time nurse in the facility are entailed to update a patient's medical records during the working days and between 8 AM and 5 PM. (3) Physicians can give all or part of their authority to another physicians while they are travelling or in the urgent unit. (4) The healthcare system is accessible in the space by physicians through hand-held devices. Thus, physician context is needed with access requests to determine what actions are permissible to be performed in the system. (5) In case of emergency, the system needs to collaborate with other agents to send an ambulance and fire-truck to provide healthcare services to patients everywhere.

The authorization challenges can be summarized as follows:

1. User population is dynamic and the identities of all users are not always known in advance. Classical RBAC is not capable for supporting systems that provide services to unknown users. This is because RBAC requires users to be authenticated for access control. When users are dynamic and their identities are unknown, determining users' authorization is challenging.

2. Some functions may have limited or periodic temporal duration. Traditional RBAC does not have temporal support. It has no provision for role activation during specified temporal intervals.

3. Users can give all or part of their authority to another user. This requirement needs support for delegation as well as revocation of delegation neither of which are porvided by traditional RBAC.

4. A user's context determines what actions can be performed in the system. Context typically is application dependent. Traditional RBAC components are limited to user, role, permission and constraint. There is no support for incorporating contextual information such as location or other environmental conditions in the model.

5. A system needs to collaborate with other systems to provide some services. RBAC is intended for a security control under a single administrative domain. Challenges that arise from collaboration, such as policy reconciliation or support for ad hoc policies, are beyond the scope of RBAC.

The central research question of this study is that *Is there a single extended RBAC model (or another authorization model) that is suitable for addressing these requirements?*. For this purpose, we investigated five recent RBAC extensions in different dimensions [2, 3, 4, 6, 11]. However, we found all these models to be inadequate in addressing these five security requirements at the same time. In the next few sections, we describe these newer extensions in more details.

4 Newer RBAC Extensions

This section discusses the current RBAC extensions that support above mentioned security requirements.

4.1 Addressing the Challenge of Un-known Users

An integrated model to RBAC called a credential-based access control has been proposed to overcome this shortcomings of RBAC and facilitate security administration [7]. In this model, a user is allowed to gain specific access privileges based on provided credentials, such as, credit card numbers or proof of membership. These credentials are used to provide information about the user's rights, qualifications, responsibilities and other attributes. Based on these information a determinatiuon is made on whether it is safe to provide access to the user.

This model has some shortcomings. Once it grants a privilege it has no way of revoking that or escalating that based on the user's behavior. It does not bind user behavior history with required privileges for each session to make access decision. TrustBAC [4] is proposed to fill this gap. Using this model, access authorization is granted based on the user's trust level.

The model evaluates trust relationships based on vector model of trust [12]. In TrustBAC model trust is always related to a particular context. For example, entity A needs to compute the trust level of entity B in some context. A trust relationship for particular context c is a vector $(A \xrightarrow{c} B)_t$ of three components: experience, knowledge, and recommendation. It is represented by $[{}_AE^c_B, {}_AK^c_B, {}_AR^c_B]$ where ${}_AE^c_B$ represents A's experience about B in context c, ${}_AK^c_B$ represents As knowledge, and ${}_AR^c_B$ represents B's recommendation to A from different source. These three factors are expressed in terms of numeric values in the range $[-1, 1] \cup \{\perp\}$. In this range 0 indicates *trust-neutral*, -1 indicates *trust-negative*, and 1 indicates *trust-positive*.

4.2 Incorportaing Temporal Information into RBAC

In temporal domains, a role has predefined activation or deactivation time. A role is get activated within the duration a role is in enabled state. Sometimes, there should be some activation dependences between roles. For example, a role called doctor-on-night-duty is active between 10 PM and 6 AM. Because a doctor needs the assistance of a nurse, there should be another role of nurse-on-night-duty in the same period. Therefore, whenever doctor-on-night-duty is active/non-active nurse-on-night-duty must be active/non-active respectively.

Temporal-RBAC (TRBAC) [6] extended RBAC model to support such temporal constraints. To support role dependencies, the concept of role triggers is defined. These role triggers executed whenever activation and/or deactivation of roles takes place. The firing of a trigger may cause an immediate or differed action of activation/deactivation of roles. Role triggers resolves conflicting roles activation/ deactivation, and an activation priority. The action of high priority is the winner and it will be executed first. Further, an administrator can issue run time requests of activation/deactivation of roles. When a user request to activate a role, the system authorizes the user if the user has the authorization to play the role and the role is enabled at the request time. Additionally, run-time requests, periodic events, and role triggers are also prioritized to solve the problem of conflict actions.

4.3 Incorporating Role Delegation into RBAC

Delegation of authority is an important business rule related to the access control policies. A delegation of permission is necessary in the following situations: (a) Role backup: The job functions of an individual need to be maintained by others when this individual is on a business trip for a long period (b) Decentralization of authority: The higher job position's function are distributed to the lower job positions, and (c) Collaboration of work: Employees need to collaborate with each other to perform job.

Researchers have developed number of models that deal with these security requirements. In particular, RBDME model [13] based on RBAC supports this requirements. Such models supports user-to-user delegation where a user in role (delegator role) grants his/her role membership to another user in another role (delegate role). This means that a delegator (a user) assigns delegetee (another user) to some role.

In many cases such a user wants to assign to another user a piece of a role's permissions which called partial delegation. However, the model of delegation in [13] is not able to cover these cases. Permission-Based Delegation Model based RBAC (PBDM) is proposed by Zhang et al. [3] to fill this gap. Using $PBDM$ model, a delegator can delegate his/her entire or partial permissions to others and partial revocation is also possible. The PBDM is a consecutive development of three models named PBDM0, PBDM1, PBDM2.

PBDM0: This model focuses on temporary delegation rather than durable delegation such that a user can delegate both permissions and roles to other users. The revocation is the undo process of the delegation. An individual can remove his/her own temporary delegation roles (DTR) at any time. Revocation is required for the three cases: (a) Remove user from delegatees, that is, revoke the user-delegation role assignment, (b) Remove one or more pieces of permissions from temporary delegation role, and (c) Revoke temporary delegation role. However, PBDM0 has two main shortcomings. First, security administrators have no control over user-to-user delegation so that a delegator might violate security policy by delegating permission or roles to malicious delegatees. Second, it does not support role-to-role delegation which is needed in many cases. PBDM1 is developed to address these problems.

PBDM1: In this model there are three layers of roles: regular role (RR), delegatable roles (DBR), and delegation roles (DTR). Permission assigned to the regular roles can be delegated to any users or roles whereas permissions assigned to the delegatable roles can be assigned to any other users or roles through delegation role. Therefore, a delegator can only delegate some of his delegatable permissions to others. Using this model, a security administrator manages the permission-regular role assignment (PAR), user-regular role assignment (UAR), permission delegatable role assignment (PAB), and user-delegatable role assignment (UAB), while individual user manages the permission-delegation role assignment (PAD) and user-delegation role assignment (UAD). PBDM0 and PBDM1 do not provide user-to-user delegation. PBDM2 is developed to fill this gap.

PBDM2: Here roles are divided into four different layers: regular roles (RR), fixed delegatable roles (FDBR), temporal delegatable roles (TDBR), and delegation roles (DTR). RR and FDBR are exactly the same as RR and DBR in PBDM1. Essentially PBDM2 is a role-to-role delegation. In this model, no individual user can own any delegation roles and permissions; instead, all delegations authority are managed by a security administrator. In PBM2, a delegator is not a user and it is a fixed delegatable role.

4.4 Incorporating User Context Information into RBAC

A central aspect of a pervasive computing application is context awareness. This is used in many applications such as hospital information systems. The information about a user's current physical location, the devices being used, the network access node, and current user activity are some examples of contexts that are integrated into an access request. However, incorporating user context information to control access to a system is challenging. First, it requires the use of sensors to acquire context information the authenticity and integrity of which must be guaranteed. Since sensors can be easily compromised, this is challenging. Second, the dynamic nature of information may require revoking role membership with a change of context information. This is challenging as it involves dynamic changes to policies during deployment. A Context-Aware Role-Based Access Control (CA-RBAC) has been proposed by Kulkarni and Tripathi [2] to support some of these requirments.CA-RBAC has two distinct operational layers – context management and access control.

The context management layer is responsible for aggregating sensor data to generate context information required by a system. For applications implementing such model trusted context agents need to be installed. The system needs to send a context query to one of the context agents for each access request. The context agents need to authenticate sensors and validate the data integrity aggregated from sensors.

The access control layer is responsible for managing context-based access to resources based on personalized permissions. The concept of personalized permissions for each role member is an extension to classic RBAC where permissions for all members of a role are always the same. With such requirement distinguishing between role permissions for different role members is mandatory for making access decision. The constraints of context-based access to resources are expressed by predicates and only those resources satisfying these predicates are allowed to be accessed through role permissions.

4.5 Incorporating System to System Interaction into RBAC

System-to-system interactions over a network such as that found in Web Services, are supported by dynamic and distributed applications in contrast to more traditional systems like centralized and client-server applications. Such applications provide local and global services to their users. The local services do not involve services from other service providers. The global services need a collaboration with other remote service provider. Sometimes, the global services might be composed from local and global services from other system provider. Several security models have been proposed to protect access to shared services [14, 15]. The problem with these models is that they do not deal with global services. CGRBAC is a model proposed by Haibo and Fan [11] that deals somewhat with composite/global services.

In CGRBAC, roles are distinguished between local roles and global roles to represent the local services and global services. Global roles include local roles and global roles from other service providers. A user can be assigned to more than one global role. Local roles have permissions to call one or more local services and global roles hold permissions to call one or more global permissions. Using CGRBAC model, a user is defined as a service requester.

5 The Challenges of a Consolidated RBAC Model

To date, there is no RBAC model that can capture all the aforementioned security requirements. Each of the discussed RBAC extensions are designed to support particular domain specific security policies. In particular, they are developed to specify at most one of the five above access control requirements. Typically, consolidating all security features in a single model remains a challenging task for security pioneers. One of the important challenges is that various features in one model might result in inconsistency and conflict deficiencies in system implementing such model due to improper features interaction. For example, one security rule may allow a user to activate a role on the satisfaction of temporal constraints associated with that role; at the same time, another security rule mayprevent that user from activating the role because user trust level is not in the role trust interval. Towards this end, such security flaw is a source of security breaches. Such security flaws can be inadvertently exercised by authorized user or purposely by an intruder resulting unauthorized access to protected resources or services. Nevertheless, over the past few decades, researchers have proposed some RBAC extensions that incorporate multiple RBAC extensions in different dimensions into one solid model. Recently, an RBAC extension capable to specify temporal and spatial constraints into one model termed as spatio-temporal model [16]. The spatio-temporal model combines the temporal aspects is TRBAC [6] and location constriatnts defined LRBAC model [17] in one model. However, the spatio-temporal RBAC suffer from a lack of formalizm. In particular, the model still separately specify the set of locations and the set of duration for role enabling, assignment, and activation in isolation. Another example from health care domain, a doctor might request a medical information of a patient from another facility provider, CGRBAC supports global dervices can partial satisfy this security requirement. It is highly possible the patient medical information is exposed by that doctor. Therefore, we need to evaluate the doctor profile of such interactions. The later security requirement is supported by TrustBAC model. However, to the best of our knowledge, there is no RBAC extension that can specify both security requirements.

Another important issue is the formal security analysis of the consolidated model to ensure the correctness of the access control policies specified by that model. Over the past decades, various frameworks for formal security analysis of RBAC are proposed in the literatures [18, 19, 20]. Such verification approaches and many others are developed with purpose of verifying a specific RBAC extended model. Despite the development of these formal verification approaches, there are still relevant analysis issues that have not been yet addressed in the literatures. Many of these approaches are primarily static in nature and do not take into into account various aspects of dynamism of RBAC. For example, verification approaches that are using model checking Coloured Petri Nets (CPN) are static in nature because the initial marking of the CPN model should be instantiated with a specific number of tokens which remain stable during the verification process. Further, some of these security analysis approaches are not automated, thus manual verification is error prone, time consuming, and cannot ensure the non-existence of flows in a security policy. Although, the automated analysis approaches fill this gap, such approaches suffer from a ever existing problem, i.e. state explosion. Specifically, an automated verification of the various features interaction, the state space and time increase drastically with the increase in the number of features. Further, some RBAC extensions have been verified yet. For example, none of the existing verification approaches considered the formal analysis of the TrustRBAC model because of the dynamic nature of the users trust level is not easy to capture.

6 Conclusion

This study have presented a multi-centric application which requires a consolidated authorization model to support the security requirements. We have investigated the state-of-art of the access control models. In particular, we have investigated the current RBAC extensions as they are most influential authorization models in the security community. The study, however, showed that, all or most of the exiting RBAC extensions are not suitable for specifying security requirements of that application.

Although this study might not be exhaustive, we believe it will be useful for researchers to be involved in investigation of RBAC models to provide another picture of the current state-of-the-art. This study indicates the need of the development of a consolidated model or provide some approaches for supporting the security requirements of the application presented in this study. The investigation in this study open the doors to the software stake holder to identify which of the investigated RBAC extension efficient or might need some configuration to support the corresponding security policies.

References

[1] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based access control models," IEEE Computer, 29(2):38-47, 1996.

[2] K. Devdatta and T. Anand, "Context-aware role-based access control in pervasive computing systems," In Proceedings of the 13^{th} ACM Symposium on Access Control Models and Technologies, Estes Park, CO, 2008

[3] Z. Xinwen, O. Sejong, and S. Ravi, "PBDM: a flexible delegation model in RBAC," In Proceedings of the 8^{th} ACM Symposium on Access Control Models and Technologies, Como, Italy, 2003.

[4] S. Chakraborty and I. Ray, "TrustBAC: integrating trust relationships into the RBAC model for access control in open systems," In Proceedings of the 11^{th} ACM Symposium on Access Control Models and Technologies. Lake Tahoe, CA, 2006.

[5] R. Sandhu. "Role hierarchies and constraints for latice-based access controls." In E. Bertino, H. Kurth, G. Martella, and E Monotolivo Eds. LNCS 1146, Proceedings of the European Symposium on Research in Computer Security 1996, Rome, Italy.

[6] E. Bertino, P. A. Bonati, and E. Ferrari, "TRBAC: A temporal role-based access control model," ACM Transactions on Information and System Security, 4(3):191-233, 2001.

[7] M. Blaze, J. Feigenbaum, and J. Ioannidis, "The KeyNote trust management system version 2.", Internet Society, Network Working Group. RFC 2704, 1999.

[8] R. Sandhu, V. Bhamidipati and Q. Munawer, "The ARBAC97 model for role-based administration of roles," ACM Transactions on Information and System Security, 2(1):105-135, February, 1999.

[9] J. D. Moffett, "Delegation of authority using domain based access rules," PhD Thesis. Department of Computing, Imperial College, University of London. 1990.

[10] J. E. Bardram, T. R. Hansen, M. Mogensen, and M. Sgaard, "Experiences from real-world deployment of context-aware technologies in a hospital environment," In Proceedings of the 8^{th} International Conference on Ubiquitous Computing, Orange County, CA, 2006.

[11] H. Shen and F. Hong, "A context-aware role-based access control model for web services," In Proceedings of the IEEE International Conference on e-Business Engineering, Beijing, China 2005.

[12] I. Ray and S. Chakraborty, "A vector model of trust for developing trustworthy dystems," In P. Samarati et al. Eds, LNCS 3193, Proceedings of the 9^{th} European Symposium of Research in Computer Security, Sophia Antipolis, France, September 2004.

[13] E. Barka and R. Sandhu, "Framework for role-based delegation models," In Proceedings of 16^{th} Annual Computer Security Application Conference, New Orleans, LA, 2000.

[14] E. Damiani, S.D.C di Vimercati, S. Paraboschi, and P. Samarati, "Fine grained access control for SOAP e-services," In Proceedings of 10^{th} International Conference on World Wide Web, Hong Kong, 2001.

[15] R. Bhatti, E. Bertino, and A. Ghafoor, "A trust-based context-aware access control model for web services," In Proceedings of the IEEE International Conference on Web Services, San Diego, CA, 2004.

[16] I. Ray and M. Toahchoodee, "A spatio temporal role based access control model," In Proceedings of the 21^{st} Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Redondo Beach, CA, 2007.

[17] I. Ray, M. Kumar, and L. Yu, " LRBAC: A location-aware role-based access control model," In Proceedings of the 2^{nd} International Conference on Information Systems Security, Kolkata, India, 2006.

[18] M. Toahchoodee and I. Ray, "On the formal analysis of a spatio-temporal role-based access control model," In Proceedings of the 22^{nd} Annual IFIP WG 11.3 Working Conference on Data and Applications Security, London, U.K., 2008.

[19] B. Shafiq, J. B. D. Joshi, and A. Ghafoor, "Petri-net model for verification of RBAC Policies," Technical Report, Purdue University, 2002.

[20] A, Schaad and J. D. Moffett, "A lightweight approach to specification and analysis of role-based access control extensions," In Proceedings of the 7^{th} ACM Symposium on Access Control Models and Technologies, Monterey, CA, USA, 2002.