Dynamic Access Control Administration for Collaborative Applications

ANDREAS MATTAS¹, IOANNIS MAVRIDIS², CHRISTOS ILIOUDIS¹ and IOANNIS PAGKALOS¹

¹Informatics Laboratory, Faculty of Technology
Aristotle University
54006, Thessaloniki
GREECE

²Department of Applied Informatics
University of Macedonia
156 Egnatia Street, 54006, Thessaloniki
GREECE

Abstract: - Today’s web-based collaborative applications need new approaches to overcome the shortcomings of classical access control. The limitations on administrative aspects of the existing security models and the requirements for more efficient management of authorizations in order to provide fine-grained and just-in-time access control for collaborative applications are discussed in this paper. The proposed DARBAC (Dynamically Administering Role Based Access Control) model provides a new approach for the administration of access control in such environments, where often the security breaks. The DARBAC model provides fine-grained and dynamic administration of authorizations in RBAC-based access control for a wide-range of collaborative applications, usually set-up in Web-based environments. An experimental implementation of the DARBAC model confirmed a number of benefits related to the administration of RBAC in collaborative environments.

Key-Words: - Access control, meta access, RBAC, DARBAC

1 Introduction

New trends in modern web application systems (e.g. tasks coordination using workflow management, collaborative computing, large-scale distribution, etc) have further complicated the task of determining the most appropriate authorization schema and have resulted in a significant increase of the overall administrative overload [1]. Access control administration (or meta-access control [2]) regulates and enforces the definition of the components of an access control system by authorized administrative users. Access control administration depends however strictly on the number of the handled components, a parameter implying the rate of simplicity of the access control system, and on the frequency of changing permissions. In general, when compared to system security, application security is characterized by an increased complexity due to the large number of variations and combinations of objects and operations to be protected [3]. This complexity increases sharply in collaborative and distributed environments, as possible subjects may vary and their actions depend on environmental or any other contextual conditions that differentiate their need-to-know requirements. In such circumstances, access control administration tends to be particularly costly and prone to error.

Access control administration seems to be basically simple as long as the sets of permissions do not change often. The complexity of administration increases however sharply in collaborative and distributed environments. According to Sandhu [4], one of the main omissions of current RBAC based models is the authorization of administration. Security models for web applications must provide efficient and dynamic administration of authorizations to ensure that valid users exercise their privileges only during the progress of an official activity of their organization. Hence, additional access control mechanisms are needed to record dynamic changes in the content and context of information and monitor the state of the system [5]. The corresponding mechanisms must avoid overloading the administrative task.

A number of important issues related to applications in collaborative environments have been recently discussed in the literature. These include ([6]) separation of duty, strict least privilege, order of events, delegation of the authority, enforcement of access control at a distributed platform level, specification of permissions based on varied information like roles and context, scalability in terms of the quantity of collaborative operations, high granularity of information and resources protection, high level specification of permission, and dynamic specification and change of policies at runtime depending on the collaboration dynamics.
Finally, another significant requirement for access control in collaborative environments is related to meta access control or access administration. Meta access control can either be incorporated within the basic access control model or provided through a separate model [2]. The DARBARC model, which is presented in the next section, can be considered as an access control model that incorporates meta access control for dynamic and distributed management of roles and permissions at runtime.

2 Limitations of Current Approaches
In today’s web applications, access control must be based on permissions available (through activated roles) just-in-time to proper users accomplishing specific tasks. Existing access control approaches have however serious limitations on this and as a result cannot cope satisfactorily with such web-based collaborative computing environments.

Pure RBAC [7] is mainly suitable for function-oriented hierarchical organization structures (vertical approach) usually used in relatively stable environments. Moreover, the use of roles for grouping authorizations in RBAC is based on general organizational terms without incorporating user, object or process attributes [6]. Current organizational alternatives often adopt however matrix organization or mission-oriented structures that lead to access control capable of embedding the required application-level contextual information. In addition, they introduce dynamic, process-based functional requirements for activity-intensive web applications constituted by a complex mixture of tasks that may transcend organizational boundaries [1]. Therefore, Web applications should be able to express application-level access control policies in new application domains, like collaborative environments that allow groups of users to cooperate on common tasks, e.g. workflow management systems.

Administration of authorizations in RBAC is centralized. However, the increasing interest of the research community for the just-in-time and active security, which is easy adaptable to particular environments, has lead to a number of other approaches as well. None of which though adequately addresses the above requirements of web-based collaborative environments.

An alternative approach to control distributing of administrative authority in a hierarchical way is offered in Administrative Role-Based Access Control (ARBAC) [8], which proposes roles to be used as a suitable mechanism for managing roles. The Task-Based Authorization Control (TBAC) and the TeaM-Based Access Control (TMAC) models are two interesting alternatives. The TBAC model [4] extends the traditional subject/object-based access control models by including domains that contain task-based contextual information. Authorizations are changed and granted in steps related to the progress of tasks. TBAC supports dynamic but centralized administration of authorizations, which is based on contextual parameters related only to task progress [6].

The TMAC model [9] also introduces the metaphor of teams, providing so a paradigm for access control that is natural to the collaborative work. In TMAC, team is considered as a general structure with role-based permission assignments to object-types. Contextual information is associated with collaborative tasks and is further applied to decision making for activation of permissions.

The Context-based TMAC (C-TMAC) model [10] is a further extension of TMAC, which integrates RBAC and TMAC concepts and uses a wider range of contextual information (time of access and location from which access is requested) that is incorporated in the overall architecture. C-TMAC preserves the advantages of scaleable security administration of RBAC model and yet offers more flexibility to control activation of permissions for individual users and specific object instances. C-TMAC extends the notions of team and context, but it lacks the fine-grained and self-administration of entities and relations, as well as the definition of multidimensional contexts [6].

Finally, the PBDM family of models [11] also addresses the issue of authority delegation, dealing with various types of delegation: user-to-user, user-to-role, role-to-role. PBDM introduces however a great deal of administrative intervention for controlling or performing those delegations in large distributed application systems. It can be said therefore that existing access control approaches cannot cope satisfactorily with modern web-based collaborative computing environments.

3 The DARBARC model
The proposed DARBARC model aims to address this issue and to provide fine-grained and dynamic administration of authorizations in RBAC-based access control for a wide-range of collaborative applications, usually set-up in Web-based environments. The DARBARC model relies upon two families of models, RBAC [7] and PBDM [11]. However, the DARBARC model awards priority to the administration of access control during run-time.
Specifically, the DARBAC model assumes a distributed Web-based information system wherein a number of missions having possible relationships between them are in progress. The protection state of the whole system is partitioned in particular states relating to each one mission instance. An overview of the DARBAC model is shown in figure 1.

3.1 DARBAC structure

The DARBAC model consists of six sets of entities called users, organization roles, mission roles, permissions, missions and objectives (Figure 1).

A user is a subject that uses a Web-application. Permissions are modes of access, which users can exercise on one or more data objects. Permissions are actually pairs, each one consisting of an object and an operation for that object. Roles used in DARBAC are distinguished between organization roles and mission roles. An organization role is a job function or title within the organization with some associated semantics regarding the authority and responsibility conferred on the user that is member of the organization role. Mission roles are application-specific and temporarily activated or delegated to other mission roles.

Mission roles are used as an intermediary between organization roles and permissions and are activated only in the frame of mission instances. Mission roles are grouped in organization roles that in turn are assigned to users. Mission roles are used in a different way, as will be demonstrated in the rest of this paper. Furthermore, mission roles are characterized as regular (RR), fixed delegatable (FR), temporal delegatable (TR) and delegation (DR) roles, according to the PBDM2 model [11]. Regular roles correspond to roles in the core RBAC [7], whereas fixed delegatable, temporal delegatable and delegation roles are used only for role-to-role delegation purposes. A fixed delegatable role owns a set of delegation roles. A temporal delegatable role can receive permissions delegated by a fixed delegatable role.

An important difference between DARBAC and other role-based approaches is that sessions are not included in its components because they are replaced by the notion of mission instances, which are more suitable for Web-application environments. A mission is defined as a type of project/process, that is carried out by a user or a team/group of users that are members of appropriate organization roles. In collaborative environments, missions are carried out by teams of users. A mission may also be entitled to a part (tasks) or an entire workflow. The structure of a mission represents a temporary project team organization, like an adhocracy organization [12]; for a finite period of time the members of a project team are acting in order to accomplish specific goals that are determined by a number of objectives. When goals are accomplished, the particular mission instance is ended.

The notion of objective is similar to context, as it was defined in [9], but is used in a broader sense. An objective may include the particular target/object of an operation; for example, the payment of a check concerns a given check that is identified by the check’s number. In general, an objective instance may contain values of time intervals, object identities, and other contextual information that contributes to specify a restricted range to exercise generally applied permissions. An objective is defined by the Cartesian product of different sets of objective types, where each objective type represents a domain of contextual information.

The following definitions, which are based on the RBAC [13] and PBDM [11] families of models, provide some formalization to the above discussion.

**Definition 1 – Entities:**

- U, P, M, stand for users, permissions and missions, respectively.
- \( O = \{2^{OT_1} \times 2^{OT_2} \times ... \times 2^{OT_n}\} \), an objective is a set of combinations of objective types \( OT_i \) from a finite set \( \{OT_1, OT_2, ..., OT_n\} \).
- \( O_i \), is an objective instance; actually an n-tuple of values of an objective; for example the objective \( \{(21:30, 22:15), (Monday, Tuesday, Friday), (127.0.0.1)\} \) that is defined for objective types time, day, IP address, respectively.
- \( M_i \), is a mission instance.
- OR, is an organization role.
- MR, is a mission role that can be further refined in regular (RR), fixed delegatable (FR), temporal delegatable (TR) or delegation (DR) role.

Permissions are assigned to mission roles with a many-to-many Permission-to-Mission Role Assignment relation (PMRA). A mission role can have many permissions and the same permission can be assigned to many mission roles. Users are also related to organization roles with a many-to-many User to Organization Role Assignment...
A user can own many organization roles, and an organization role can be assigned to many users. Furthermore, mission roles are assigned to organization roles with a many-to-many Mission role to Organization role Assignment relation (OMRA). Objectives are assigned to missions with a many-to-many Objective to Mission Assignment relation (OMA). The set of skills needed for a mission to accomplish its goal is specified through an Organization Role to Mission Assignment relation (ORMA), where each role refers to an autonomous activity of mission, giving an alternative way to represent tasks in workflows.

During run-time, users with sufficient administrative permissions can bind an objective instance to a mission instance through a many-to-one Objective instance to Mission instance Binding relation (OMB). In addition, when a user participates in a mission instance, a new entry is added in a many-to-many User to Mission instance Participation relation (UMP).

**Definition 2 – Relationships:**
- PMRA ⊆ P × MR, is a many-to-many permission to mission role assignment relation.
- UORA ⊆ U × OR, is a many-to-many user to organization role assignment relation.
- OMRA ⊆ OR × MR, is a many-to-many organization role to mission role assignment relation.
- UMP ⊆ U × Mi, is a many-to-many user to mission instance participation relation.
- OMB ⊆ Oi × Mi, is a many-to-one objective instance to mission instance binding relation.
- OMA ⊆ O × Mi, is a many-to-many objective to mission assignment relation.
- ORMA ⊆ OR × M, is a many-to-many organization role to mission role assignment relation.

Two types of constraints can be defined by security designers during build-time: Separation of Duty Constraints (SDC) and Join of Duty Constraints (JDC). SDC and JDC constraints are enforced during run-time to determine users’ participation in mission instances. SDC imposes the rule that no user can participate in a mission instance with more organization roles than the specified ones. JDC rules that a user, who is a member of a given organization role, can participate in a mission instance, subject to participation of a second user, who is a member of another organization role. In addition, identity-based inclusions/exclusions of users to participate in a mission instance can be specified by administrative or regular users during run-time with User-Mission Constraints (UMC); for example, a patient that controls accessing to his health-care record.

**Definition 3 – Constraints:**
SDC is expressed as a set of pairs (rs, n). A set rs includes the organization roles, that are assigned to missions (as defined in ORMA) and also are in separation of duties. SDC imposes the rule that no one user can participate in a mission instance with n or more organization roles from the set rs.

Let SDCm be the set of pair (rs, n) ⊆ (2^R × N) that is defined for mission m ∈ M, where rs ⊆ {r ∈ OR | (r, m) ∈ ORMA}, n ∈ N, |rs| ≥ n ≥ 2. Let also sr_u be a subset of organization roles in rs (sr_u ⊆ rs) that the same user may activate when participating in the same instance of mission m ∈ M. Then, the cardinality of sr_u (the number of organization roles that the user can activate to participate in the same instance of mission m), cannot exceed the value of n: ∀ (rs, n) : SDCm, ∀ sr_u ⊆ rs • |sr_u| < n

JDC is expressed as a set of triples (r1, r2, ⊕), where r1, r2 ∈ OR and ⊕ is a symbol meaning that a user with organization role r2 can participate in a mission instance, only if another user with organization role r1:
- is already participating, regardless when the user with organization role r1 will stop participating in the same mission instance (⊕ is replaced by symbol >), or
- is not concurrently participating (⊕ is replaced by symbol ≠), or
- is concurrently participating (⊕ is replaced by symbol ||)

Let JDCm be the set of triples defined for mission m ∈ M. Let also DT_ri be the period of participation of a user with role r_i ∈ OR, i=1, 2… N, in an instance of mission m. Then, for any instance of mission m, each one replacement of symbol ⊕ is defined as follows:
∀ (r1, r2, >) : JDCm • DT_r1 ∩ DT_r2 ⊆ DT_r1
∀ (r1, r2, ≠) : JDCm • DT_r1 ∩ DT_r2 = Ø
∀ (r1, r2, ||) : JDCm • DT_r1 ∩ DT_r2 = DT_r2

UMC: By default, all users with appropriate organization roles can participate in an instance mi of mission m ∈ M if constraints SDCm and JDCm are satisfied, unless specified differently in a constraint UMCm.

Let UMCm be the set of pairs (u, ⊗), specified for instance mi of mission m ∈ M, where u ∈ U and ⊗ is a symbol. Then, a user u cannot participate in mission instance mi, if:
- an exclusion has been specified for that user (⊗ is replaced by symbol −), or
- inclusions have been specified for other users except user u (⊗ is replaced by symbol +).
3.2 DARBAC processes
The definition and managing of the components of a DARBAC-based access control system is an administrative process with two phases [14]: build-time and run-time.

During build-time, the organization managers in cooperation with the application developers capture the organization's rules and policies to define, name and construct the components of the access control system: missions, objectives and their assignment relationships, organization roles and mission roles.

Access control managing during run-time copes with situations related to any daily or emergency conditions in an organization. Administrative operations are performed by the administrators, as well as regular users under certain constraints.

In the presence of a user access request, the DARBAC decision-making process is based on the permissions the user acquires through mission roles and the particular values of the objective instances bound to the mission instance. The decision-making process is accomplished in a four-step procedure (as depicted in Figure 2):

**Step 1: Reviewing**
User u activates either manually or automatically (through the application) the proper organization role to participate in mission instance $m_i$. In order to perform an operation, he uses the Web-application to submit the appropriate access request along with the name of the activated organization role to the access control system. Then, given the bound objective instance, the mission instance $m_i$ is sought in the relation OMB. In the case the mission instance $m_i$ is not found, the access request is denied. Otherwise, the relation UMP is sought for a previous participation of user u in the same mission instance. If such an entry is found, the procedure goes to step 3; otherwise it continues to step 2.

**Step 2: Participating**
The user’s participation in the mission instance is initially determined, on a base of static information, according to the following sub-steps:
- Relation UORA is checked to confirm that the organization role presented by the user u has been assigned to him. Otherwise, the user’s access request is denied and the procedure ends.
- Relation ORMA is sought for appropriate entries that permit a user to participate in an instance of mission $m$, after he has activated the presented organization role.

However, no change or addition of a new record in UMP is provoked due to the need for verifying the participation of user u under the current context of mission instance progression.

**Step 3: Checking**
The dynamics associated with the current state of the access control system, as they are expressed by the constraints SDC$m$, JDC$m$ and UMC$m_i$, are examined to verify the user’s participation in the mission instance, as follows:
- The constraints SDC$m$, JDC$m$ and UMC$m_i$ are applied on the current values of names of user and organization role.
- In the case the above constraints are satisfied, the user u is allowed to participate in the mission instance $m_i$ and the relation UMP is updated accordingly.

Besides the fact that the user u already participates in the mission instance $m_i$, this step has to be repeated for any subsequent user’s access request in order to verify his participation under the current dynamically changing conditions.

**Step 4: Matching**
The final judgment of whether the user has the right to execute his access request or not is taken according to the results of the following actions:
- Aggregation of user’s permissions, based on the mission roles either assigned to the organization role (regular roles) during build-time or delegated (delegation roles) during run-time, and
- Matching the requested operation to the acquired permissions.

4. Implementation
An experimental implementation of the DARBAC model that comprises the development of a DARBAC-based meta-access and access control system and its application in the banking sector has been carried out [15]. For demonstration purposes, a typical workflow transaction of check payment has been used. The implementation assumes Web-based
applications to support enforcing of access control at a distributed platform level, and demonstrates in a step-by-step basis the construction of DARBAC components and their management during run-time. The underlying software platform included Windows Server 2003 with Internet Information Services (IIS), Microsoft ASP.NET, and Microsoft .NET Framework for Web-enabled user interface.

6 Discussion - Conclusion
In this paper we proposed a new access control approach that preserves the advantages of permission administration that RBAC models offer. Moreover, the DARBAC model introduces the concept of mission, in addition to that of roles. Mission is used as an abstract mechanism to formulate the objective information, which in turn identifies the aim and context of activities to be performed by a group/team of users with prerequisite roles. Dynamic administration is achieved by granting to normal and administrative users the proper administrative permissions to manage missions and their objectives, to perform temporary delegations and to apply a number of constraints that concern the functional dependencies governing the participation of users in missions. In addition, a distinction is made between build-time operations for static security analysis and design, and run-time operations for dynamic checks on the security aspects of the system to support a rich set of security policies that tune permission management in flexible ways. The proposed DARBAC model provides, amongst others, security features that comprise temporal role activation, controlled decentralization of administrative care, constraint-based privacy protection, dynamic separation of duties, and synchronization of permission availability for users with different responsibilities.

References:
[12] Shim, W.B., Park, S.; Toward an Improved RBAC Model for the Organic Organization, 9th International Conference on Parallel and Distributed Systems, Taiwan, 2002