Role-based Access Control

Prof.Dr. Ferucio Laurenţiu Ţiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iaşi
Iaşi, Romania
E-mail: fltiplea@mail.dntis.ro

Outline

1. Introduction to RBAC
2. Basic RBAC
3. Hierarchical RBAC
4. Constrained RBAC
5. Consolidated RBAC
6. RBAC Implementations
7. Concluding Remarks on MAC Models

Role-based Access Control

Basic features:

- Access rights are grouped according to a particular functionality into a role
- User flexibility: a user moving to a new function is simply assigned to the new role and removed from the old one
- Powerful mechanism to an administrator to specify the privileges required by various job functions

RBAC models:

- Basic RBAC
- Hierarchical RBAC
- Constrained RBAC
- Consolidated RBAC

Users, Roles, Permissions

Basic elements:

- U is set users
- R is set of roles
- $P \subseteq \mathscr{P}(Op \times O)$ is set of permissions
  (Op is the set of operations, and O is the set of objects)
- $UR \subseteq U \times R$ is the user-to-role assignment relation
- $PR \subseteq P \times R$ is the permission-to-role assignment relation
- $su : S \to U$ is the subject-to-user mapping
  (S is the set of subjects)
- $sr : S \to \mathscr{P}(R)$ is the subject-to-role mapping, constrained by $sr(s) \subseteq UR(su(s))$

Basic RBAC

Role authorization: a subject can never have an active role that is not authorized for its user

$$(\forall s \in S)(\forall r \in R)(r \in sr(s) \;\Rightarrow\; su(s) \in UR^{-1}(r))$$

Object access authorization: A subject s can perform an operation op on object o only if there exists a role r that is included in the subject's active role set and there exists a permission that is assigned to r such that the permission authorizes the performance of op on o

$$access(s,op,o) \;\Rightarrow\; (\exists r \in R)(\exists p \in P)(r \in sr(s) \;\wedge\; p \in PR^{-1}(r) \;\wedge\; (op,o) \in p)$$

Hierarchical RBAC

Hierarchical RBAC builds on top of basic RBAC by adding a role inheritance relation which is a partial order relation $\geq$ on R

Meaning of $r_1 \geq r_2$:

- $PR^{-1}(r_2) \subseteq PR^{-1}(r_1)$
- $UR^{-1}(r_1) \subseteq UR^{-1}(r_2)$

Authorized users and permissions:

- the set of authorized users for the role r is

$$\{u \in U \mid (u,r') \in UR \ \wedge \ r' \geq r\}$$

- the set of authorized permissions for the role r is

$$\{p \in P \mid (p,r') \in PR \ \wedge \ r' \geq r\}$$

Constrained RBAC

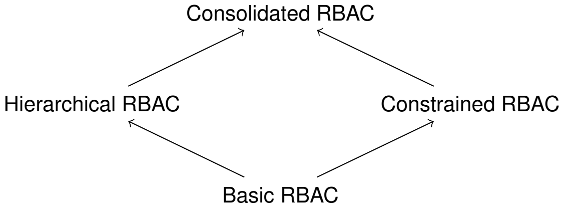
Constrained RBAC is obtained from basic RBAC by adding constraints

Types of constraints:

- Mutually exclusive roles: this is one of the most common constraint
- Cardinality: maximum number of members in some roles
- Prerequisite roles: a user is assigned to role r only if the user is already assigned to some role r' (r' proves the competency and appropriateness of the user for role r)

Consolidated RBAC

Consolidated RBAC combines hierarchical and constrained RBAC



RBAC Implementations

Implementation in two important classes of commercial software:

- Database management systems (such as Oracle enterprise server)
- Enterprise security administration

Concluding Remarks

- RBAC simplifies security administration by using roles, hierarchies, and constraints
- RBAC reduces costs within an organization because it takes into account that employees change much more frequently than the duties within positions
- RBAC can be configured to support a large variety of access control policies, including DAC and MAC policies
- RBAC is suited to a large variety of applications and software system environments