Role-Based Access Control Features in Commercial Database Management Systems

Chandramouli Ramaswamy and *Ravi Sandhu*

Computer Security Division, ITL
NIST, Gaithersburg, Maryland 20899
chandramouli@csmes.ncsl.nist.gov

Info. and Software Engg. Dept., MS 4A4
George Mason University, Fairfax, VA 22030
sandhu@isse.gmu.edu, www.list.gmu.edu

ABSTRACT This paper analyzes and compares role-based access control (RBAC) features supported in the most recent versions of three popular commercial database management systems: Informix Online Dynamic Server Version 7.2, Oracle Enterprise Server Version 8.0 and Sybase Adaptive Server Release 11.5. We categorize RBAC features under three broad areas: user role assignment, support for role relationships and constraints, and assignable privileges. Our finding is that these products provide a sound basis for implementing the basic features of RBAC, although there are significant differences. In particular, Informix restricts users to a single active role at any time, while Oracle and Sybase allow multiple roles to be activated simultaneously as per the user's selection. All three provide support for role hierarchies, but Sybase is the only one to directly support mutual exclusion of roles.

1 Introduction

Role-based access control (RBAC) has recently received considerable attention as a promising alternative to traditional discretionary and mandatory access controls (see, for example, [FK92, FCK95, Gui95, MD94, HDT95, NO95, SCFY96]). In RBAC permissions are associated with roles, and users are made members of appropriate roles thereby acquiring the roles' permissions. This greatly simplifies management of permissions. Roles are created for the various job functions in an organization and users are assigned roles based on their responsibilities and qualifications. Users can be easily reassigned from one role to another. Roles can be granted new permissions as new applications and systems are incorporated, and permissions can be revoked from roles as needed. Role-role relationships can be established to lay out broad policy objectives.

There are several mainstream commercially available products that support RBAC in some form. More than any other commercial application software, DBMSs (database management systems) provide access control at several levels of granularity including provision for content-dependent controls [San94]. An application system developed using a DBMS can contain a large amount of data with highly differentiated access permissions for different users depending upon their function or role within the organization. Hence database management is a prime area which needs a mechanisms for management of authorizations or privileges. Not surprisingly DBMSs have taken the lead in providing support for RBAC.

In this paper, we analyze and compare RBAC features implemented in the following commercial DBMSs.

- INFORMIX Online Dynamic Server Version 7.2
- Sybase Adaptive Server release 11.5
- Oracle Enterprise Server Version 8.0

In comparing the features of such complex commercial software packages, it is not always possible to readily obtain the total set of all supported features from product manuals alone. However, it is

possible to extract and compare the major differences in features from using the multiple manuals that come with the product. This is the approach that has been adopted by the authors of this paper. Our description of the feature in these products should not be construed as a complete overview. Rather we have focussed on significant issues and differences from the point of view of security administrators and developers of applications that have significant security requirements.

The features supported in INFORMIX DBMS have been obtained from [Inf97a] and [Inf97b]. For ORACLE manuals [Ora97a] and [Ora97b] were used and for SYBASE their system guides [Syb97a] and [Syb97b] formed the source of reference.[1]

The RBAC features that are supported have been categorized under three broad areas as follows.

- User role assignment
- Support for role relationships and Constraints
- Assignable privileges

We adopt the terminology and concepts of the well-known RBAC96 family of models due to Sandhu et al [SCFY96, San97].

In the next three sections we present the RBAC features of each product individually followed by a comparative discussion of all three.

2 Informix Online Dynamic Server Version 7.2

User role assignment

A role can be granted to a single user, a role, a list of users or—by using the keyword PUBLIC—to all users. A user can be granted more than one role. Users who have been granted a role with the GRANT OPTION can further grant that role or delete it by using the DROP ROLE command.

A user can have only one role active at any point in time. Initially all users are assigned the role NULL or NONE, by default, when they sign on to a database. The user can enable an authorized role by means of the SET ROLE statement. The SET ROLE statement allows for specifying only one role, so the user can enable only one role at a time. Moreover, if a user executes the SET ROLE statement after a role is already set, the new role replaces the old role. This implies that a user can be active in one and only role at every moment. Informix provides no feature to specify a default active role, different from NULL or NONE, for a user.

Support for role relationships and constraints

As already stated, users who have been granted a role with GRANT OPTION as well as DBAs can grant a role to another role. This feature enables building nested roles, so it is possible to build a role hierarchy.

Informix has no features to specify mutually exclusive roles, that is sets of roles which cannot be granted to the same user. Hence it does not support static separation of duty. There is also no support for cardinality constraint to restrict the maximum or minimum number of users that can be authorized for a role. Informix does in a sense support dynamic separation of duties, that is specification of roles that cannot be simultaneously activated. However, this is more a side effect of the fact that only role can be activated at a time rather than a independent feature in its own right.

[1] The scope of this paper is limited to single databases. In particular, Oracle 8.0 comes with an add on security product called Oracle Security Server which allows global users and roles to be defined for use across multiple databases. This feature is not discussed in this paper.

Assignable privileges

Informix divides the universe of all privileges that can be assigned into three categories: database-level privileges, table-level privileges and execute privilege.

Database-level privileges refer to privileges needed to connect to a database, add new objects and perform administrative functions like security management (including transfer of object ownerships) and space management etc. They include the CONNECT privilege (ability to establish the user context to a database schema so that the user can query and modify the objects in the schema depending upon the permissions and ownerships), RESOURCE privilege (ability to create new objects in a database schema like tables, indexes and procedures) and DBA privilege (grant privileges to another user or role, create new objects under a designated ownership—the default owner of a database object is the one who created it, update rows of system catalog tables and control the growth of physical spaces by altering extent sizes etc).

Table-level privileges refer to privileges that can be granted on a base table. They include INSERT, DELETE and ALTER that are applicable for the table as a whole, SELECT and UPDATE privileges that can be selectively applied on one or more columns of a table, as well as REFERENCES (ability to reference one or more columns in referential constraints) and INDEX (ability to create permanent indexes). Privileges that can be granted on a view are SELECT, INSERT, DELETE and UPDATE. The last three privileges are only applicable if the view meets all the requirements for updating (updatable view). ALTER, REFERENCES and INDEX privileges cannot be granted on a view.

The EXECUTE privilege is applicable only for database stored procedures. It is a single privilege representing the ability to execute the stored procedure.

Informix allows only the Table-level privileges and the EXECUTE privilege to be granted to roles. Database-level privileges cannot be granted to roles.

The DBA and the owner of a database object can grant privileges to a role and can revoke that privilege later on. Informix has a AS GRANTOR clause in the statement that grants privileges to roles. Using this it is possible to designate someone else as the grantor of the specified privilege to a role. However the person who originally executed the grant statement with AS GRANTOR option can no longer revoke that privilege from the role. It is also interesting to note that the user who has been granted a role WITH GRANT OPTION can also revoke privileges from a role.

3 SYBASE Adaptive Server Release 11.5

The Sybase Adaptive Server comes with a set of pre-defined roles called system roles. The roles created for the purpose of access control in the various databases (where each database has been created for supporting one or more applications) in the Adaptive server are called user-defined roles.

The three system roles are as follows.

- sa-role (System Administrator) - for managing and maintaining all databases in the Sybase Adaptive server as well as managing and controlling the physical resources of the server
- sso-role (System Security Officer) - for performing all security-related tasks such as creation of user-defined roles and granting them to users, groups of users or other roles
- oper-role (Operator) - backup and load databases server-wide

User role assignment

A role can be granted to one or more users and any user can be granted more than one role. Granting of roles to users can be done only by System Security Officer (i.e. person with sso-role). Further, user-defined roles cannot be granted with GRANT OPTION. Hence in Sybase it is not possible for a user who has been granted a user-defined role to propagate that role to other users. This combined with the fact there will be only a handful of server logins with sso-role (mainly security administrators) results in stronger control over role assignments and proliferation.

Sybase allows users to activate multiple roles in a user session. Although the SET ROLE statement (the one that activates a role for a user) allows for specifying only one role, by repeated invocation of this statement the user is able to activate multiple roles from the set of roles that have been authorized or granted for that user. The activation process is required only for user-defined roles. Sybase system roles are automatically activated (if they do not have passwords associated with them).

The usage of Sybase's SET ROLE statement is illustrated through the following examples:

Example 1: To activate the accountant role

SET ROLE accountant WITH PASSWD "wizkid" ON

Example 2: To de-activate the auditor role

SET ROLE auditor OFF

It is also possible for the user to set up a default list of roles to be activated at the time of user login. However roles that have passwords associated with it, cannot be part of this default list. The user has to individually activate those using SET ROLE command.

The default list of active roles for a user can be built by repeated invocation of the Sybase's system procedure SP_MODIFYLOGIN. It is also possible to delete any role from the default active role set by using the same procedure. These features are illustrated through the following examples.

Example 3: To add the accountant role as one of the default active roles for jsmith

SP_MODIFYLOGIN jsmith "add default role" accountant

Example 4: To delete auditor role as one of the default active roles for susan

SP_MODIFYLOGIN susan "drop default role" auditor

Support for role relationships and constraints

Just as in Informix, a role that is created in Sybase can be granted to other roles and hence a role hierarchy can be implemented. In addition, Sybase supports a powerful feature of RBAC. This is the ability to define mutual exclusivity of roles. There are two types of mutual exclusion that have been defined.

- Two roles are in *static exclusion* if a user cannot be granted both roles.
- Two roles are in *dynamic exclusion* if a user cannot activate or enable both roles at the same time.

In this way Sybase RBAC implementation provides for enforcement of static and dynamic separation of duty policies.

Since the maximum and minimum number of users that can be assigned to a role are not attributes of the role, it is clear that Sybase does not support role cardinality constraints. However limits can be imposed on the number of roles in which a user can be active at any point in time, and the total number of roles that can be defined for the Server as a whole.

Assignable privileges

In Sybase privileges are categorized as object access permissions and object creation permissions.

Object access permissions regulate the use of certain commands that access certain database objects. They include the SELECT,UPDATE, INSERT and DELETE commands on Tables and Views, SELECT, UPDATE and REFERENCES on Tables and Columns and EXECUTE command on Stored Procedures.

Object creation permissions regulate the use of commands that create objects. These objects include databases, tables, views, rules and stored procedures.

Both categories of privileges can be granted to roles. However, object creation permissions cannot be granted with GRANT OPTION. This implies that the privileges to create new database objects cannot be propagated to other roles or users. It is also interesting to note while the ability to grant a role to users or other roles is centrally vested with the System Security Officer (the member of the sso_role), the database object owners have the ability to assign privileges to roles for objects they own.

4 ORACLE-Enterprise Server Version 8.0

User role assignment

Just like Sybase and Informix, Oracle[2] supports a many-to-many relationship between users and roles. By using the keyword PUBLIC in the Oracle's GRANT statement (which grants roles to users) it is possible to grant a role to all users with a single statement. Users who have been granted a role with the ADMIN OPTION can grant that role to other users or roles. In addition, the role grantee can also alter or drop the role. This feature is similar to Informix.

A user who has been granted one or more roles has to invoke the SET ROLE command to enable or disable roles for the current user session. If the role has a password, you must also specify the password to enable the role by using the IDENTIFIED BY clause. Unlike the other two DBMS implementations , you can specify more than one role (i.e a list of roles) in the Oracle's SET ROLE statement. In addition, Oracle has two other variations of this SET ROLE statement which gives additional flexibility in role activation as follows.

- By using the keyword ALL, the user can activate all authorized roles. This ALL clause in turn has another subclass called EXCEPT clause. By using this clause, the user can selectively exclude some authorized roles from getting activated for the session. There is however, the restriction that the roles listed in the EXCEPT clause should be (i) roles that are directly granted to the user and not inherited through role-to-role assignments, and (ii) roles that do not have passwords.
- By using the NONE clause the user can deactivate or disable all the roles for the current session.

[2] As mentioned earlier, the scope of this paper is limited to single databases. In particular, Oracle 8.0 comes with an add on security product called Oracle Security Server which allows global users and roles to be defined for use across multiple databases. This feature is not discussed in this paper.

Here are a few examples of the usage of Oracle's SET ROLE command.

Example 1: To activate roles accountant and financial_analyst each having a password

SET ROLE accountant IDENTIFIED BY acct, financial_analyst IDENTIFIED BY final

Example 2: To activate all roles except auditor

SET ROLE ALL EXCEPT auditor

Example 3: To disable all roles granted to you for the current session, issue the following statement:

SET ROLE NONE

Oracle like Sybase provides a facility to set up a default list of roles to be activated at the time of user login. This facility is enabled through the use of DEFAULT ROLE clause of the ALTER USER command. The ALTER USER command with DEFAULT ROLE clause has the same variations that are available with the SET ROLE command. This implies that (a) it is possible to specify a list of default roles to be activated (b) include all authorized roles in the default active role set except some selected roles and (c) make the default active role set null.

The following are some examples of the usage of ALTER USER command with DEFAULT ROLE clause.

Example 4: To make accountant and analyst roles as the default active roles for Scott

ALTER USER Scott DEFAULT ROLE accountant, analyst

Example 5: To make all authorized roles for Scott part of his default active role set except the auditor role

ALTER USER Scott DEFAULT ROLE ALL EXCEPT auditor;

Example 6: To remove all roles from Scott's default active role set

ALTER USER Scott DEFAULT ROLE NONE;

Just as in Sybase, roles that have passwords associated with it cannot be made members of this default active role set and have to be activated using the SET ROLE command.

Support for role relationships constraints

Similar to Informix, Oracle provides for granting a role to another and hence it is possible to create a role hierarchy. However it is not possible to define additional constraints or relationships among roles in a declarative fashion. Hence as is the case with roles in Informix, Oracle roles do not directly support enforcement of separation of duties. Similarly as in Informix and Sybase, it is not possible to specify cardinality rules for membership in roles.

Assignable privileges

Oracle privileges are categorized as system privileges and object privileges.

System privileges are rights to execute various types of commands like CREATE SESSION, CREATE TABLE etc. System privileges are not specific to a named database object or structure.

Item	Feature	Informix	Sybase	Oracle
1	Ability for a role grantee to grant that role to other users	Yes	No	Yes
2	Multiple active roles for a user session	No	Yes	Yes
3	Specify a default active role set for a user session	No	Yes	Yes
4	Build a role hierarchy	Yes	Yes	Yes
5	Specify static separation of duty constraints on roles	No	Yes	No
6	Specify dynamic separation of duty constraints on roles	(Yes)	Yes	No
7	Specify maximum or minimum cardinality for role memberships	No	No	No
8	Grant DBMS System Privileges to a Role	No	Yes	Yes
9	Grant DBMS Object Privileges to a Role	Yes	Yes	Yes

For item 6, Informix supports dynamic separation of duties as a side effect of item 2

Features 1, 2 and 3 pertain to **user role assignment**
Features 4, 5, 6 and 7 pertain to **support for role relationships and constraints**
Features 8 and 9 pertain to **assignable privileges**

Table 1: Summary

They are specific to a particular operation or class of operations on a type of object or structure (another example is SELECT ANY TABLE).

Object privileges allows users to perform a particular action on a specific table, view, sequence or stored procedure. They include the SELECT, UPDATE, INSERT, DELETE operations on Tables and Views, ALTER, CREATE INDEX operations on Tables alone, and EXECUTE operation on procedures and functions.

Both categories of privileges can be granted to roles. System Privileges can only be granted by the DBA or by a user who has been granted that privilege with the ADMIN OPTION. Object Privileges can only be granted to roles either by owner of the object or by a user who has been granted that privilege with the GRANT OPTION.

5 Discussion and Conclusion

A summary of role based access control features that are supported or not supported in the three DBMS products studied in this paper, is given in table 1.

In the area of user role assignment, we find that in Sybase, the task of assigning roles to users is a centralized activity that can be only performed by the System Security Officer and there is no feature for assigning that right to the role grantee. In Informix and Oracle, it is possible to implement this as a discretionary access control mechanism by enabling the role grantee to grant that role to other users. Thus Sybase maintains tighter control over user-role assignment.

While Sybase and Oracle provide for multiple roles to be activated in a user session, Informix

has provision for only one role to be active. Since Informix does support role hierarchies it can be argued that by suitable definition of roles and by use of role-to-role assignments, this limitation can be overcome. However, this would require anticipation of role combinations that users would like to activate and definition of a senior role in the hierarchy which combines these together in one.

An important component of security policy in many commercial environments is separation of duty. RBAC provides a conceptual framework for implementing this policy. Out of the three DBMS products we have reviewed, we find that only Sybase has directly provides features for implementation of this policy. Sybase supports static and dynamic separation of duties. Informix does in a sense support dynamic separation of duties. However, this is more a side effect of the fact that only role can be activated at a time rather than a independent feature in its own right. Hence the qualified entry in table 1, item 6.

In the area of Assignable Privileges, Informix restricts the privileges that can be granted to the roles to only DBMS Object-level privileges. In Sybase and Oracle both System-level and Object-level privileges can be granted to roles.

In summary, we find that Sybase and Oracle provide more features than Informix in the areas of user role assignment and assignable privileges. Direct support for mutual exclusivity of roles is implemented only in Sybase. Overall our conclusion is that these products provide a sound basis for implementing the basic features of RBAC, although there are significant differences.

Finally, we reiterate that the scope of this paper is limited to single databases. In particluar, Oracle 8.0 comes with an add on security product called Oracle Security Server which allows global users and roles to be defined for use across multiple databases. This feature is not discussed in this paper.

References

[FCK95] David Ferraiolo, Janet Cugini, and Richard Kuhn. Role-based access control (RBAC): Features and motivations. In *Proceedings of 11th Annual Computer Security Application Conference*, pages 241–48, New Orleans, LA, December 11-15 1995.

[FK92] David Ferraiolo and Richard Kuhn. Role-based access controls. In *Proceedings of 15th NIST-NCSC National Computer Security Conference*, pages 554–563, Baltimore, MD, October 13-16 1992.

[Gui95] Luigi Guiri. A new model for role-based access control. In *Proceedings of 11th Annual Computer Security Application Conference*, pages 249–255, New Orleans, LA, December 11-15 1995.

[HDT95] M.-Y. Hu, S.A. Demurjian, and T.C. Ting. User-role based security in the ADAM object-oriented design and analyses environment. In J. Biskup, M. Morgernstern, and C. Landwehr, editors, *Database Security VIII: Status and Prospects*. North-Holland, 1995.

[Inf97a] Informix. *INFORMIX - Online Dynamic Server - Administrator's Guide - Version 7.2, Vol 1 and 2*, 1997.

[Inf97b] Informix. *INFORMIX Guide to SQL : Tutorial*, 1997.

[MD94] Imtiaz Mohammed and David M. Dilts. Design for dynamic user-role-based security. *Computers & Security*, 13(8):661–671, 1994.

[NO95] Matunda Nyanchama and Sylvia Osborn. Access rights administration in role-based security systems. In J. Biskup, M. Morgernstern, and C. Landwehr, editors, *Database Security VIII: Status and Prospects*. North-Holland, 1995.

[Ora97a] Oracle. *Oracle 8 Enterprise Edition - Server Administrator's Guide*, 1997.

[Ora97b] Oracle. *Oracle 8 Enterprise Edition - SQL Reference*, 1997.

[San94] Ravi Sandhu. Relational database access controls using SQL. In Zella A. Ruthberg and Hal F. Tipton, editors, *Handbook of Information Security Management (1994-95 Yearbook)*, pages 145–160. Auerbach Publishers, 1994.

[San97] Ravi Sandhu. Rationale for the RBAC96 family of access control models. In *Proceedings of the 1st ACM Workshop on Role-Based Access Control*. ACM, 1997.

[SCFY96] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, February 1996.

[Syb97a] Sybase. *Sybase Adaptive Server Enterprise Security Administration*, 1997.

[Syb97b] Sybase. *Sybase Adaptive Server Enterprise Security Features User's Guide*, 1997.