Implementing Database Roles in the Enterprise Geodatabase

Jim McAbee
Agenda

- Roles Overview
- Roles and OS Groups
- RDBMS Differences
- Privilege Assignment Hierarchy
IT Security – Many Levels

Diagram showing the flow from Client to Server with various security layers:
- ArcInfo
- ArcEditor
- ArcView
- Engine
- ArcObjects
- Native Authentication
- LDAP/Central User Repository
- Firewall
- SSL
- IPSEC
- Intrusion Detection—Network
- Data File Encryption
- Intrusion Detection—Application
- RDBMS Privileges
- Row Level Security

The diagram highlights the 3rd Party component and mentions ArcSDE at the server level.
Authentication Methods and Authorization

• Authentication vs. Authorization
  - Authentication – “who is allowed in”
    “Authentication is the process by which a system verifies a user's identity”
  - Authorization or Privileges – “what they can do”
    “Authorization indicates which database operations that user can perform, and which data objects that user can access and/or manipulate.”

• Authentication Methods
  - Database
  - External – Local OS, Domain, other (e.g. LDAP, etc.)
    - Cross-OS possible typically but complex

• Authorization or Privileges
  - Object Creation (DDL – Data Definition)
  - Object Manipulation (DML – Data Manipulation)
Users - Considerations

• User Types
  - System
  - General vs. Specific (“head-less” vs. employee)
    - Editor, Viewer (service specific?), Departmental, Operations, etc….

• System Roles
  - Public
  - other

• Locked/Unlocked accounts
  - inactivity

• Password Timeout
  - automatic
  - organization policy

• User or Role based resource management
  - space, cpu, etc…
Database Architecture and Authorization Differences

- Single vs. Multiple Database per Instance Architectures
- Instance vs. Database level privileges and roles

Oracle

- Instance
  - Database
    - schema
    - schema
    - schema
    - schema

SQL Server, DB2, Postgres

- Instance
  - Database
    - schema
    - schema
    - schema
    - schema
    - schema
    - schema
Roles Overview

• Managing and controlling privileges is easier when you use roles, which are named groups of related privileges that you grant as a group to users or other roles.
• Roles facilitate the granting of multiple privileges or roles to users.
• Similar to groups in the operating system.
• Privileges can be granted explicitly to a user or via a role.
Types of Roles

- Instance vs. Database level
Various types of Roles

- **Application**
  - type of application
  - application resource usage – load

- **Functional**
  - editors vs. viewers

- **Departmental**
  - water vs. planning, new york vs. california
SQL Server example: Fixed roles

- Many RDBMS have predefined roles used to simplify administration

- SQL Server Fixed server roles
  - Used to manage instance level permissions
  - sysadmin has full administrative privileges

- SQL Server Fixed database roles
  - Used to manage database level permissions
  - Create user-defined roles for more flexibility
# Oracle Fixed Roles Example

<table>
<thead>
<tr>
<th>Role</th>
<th>Authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADM_PARALLEL_EXECUTE_TASK</td>
<td>NO</td>
</tr>
<tr>
<td>APEX_ADMINISTRATOR_ROLE</td>
<td>NO</td>
</tr>
<tr>
<td>AQ_ADMINISTRATOR_ROLE</td>
<td>NO</td>
</tr>
<tr>
<td>AQ_USER_ROLE</td>
<td>NO</td>
</tr>
<tr>
<td>AUTHENTICATEDUSER</td>
<td>NO</td>
</tr>
<tr>
<td>CONNECT</td>
<td>NO</td>
</tr>
<tr>
<td>CSW_USR_ROLE</td>
<td>YES</td>
</tr>
<tr>
<td>CTXAPP</td>
<td>NO</td>
</tr>
<tr>
<td>CWM_USER</td>
<td>NO</td>
</tr>
<tr>
<td>DATAPUMP_EXP_FULL_DATABASE</td>
<td>NO</td>
</tr>
<tr>
<td>DATAPUMP_IMP_FULL_DATABASE</td>
<td>NO</td>
</tr>
<tr>
<td>DBA</td>
<td>NO</td>
</tr>
<tr>
<td>DBFS_ROLE</td>
<td>NO</td>
</tr>
<tr>
<td>DELETE_CATALOG_ROLE</td>
<td>NO</td>
</tr>
<tr>
<td>ejbclient</td>
<td>NO</td>
</tr>
<tr>
<td>EXECUTE_CATALOG_ROLE</td>
<td>NO</td>
</tr>
<tr>
<td>EXP_FULL_DATABASE</td>
<td>NO</td>
</tr>
<tr>
<td>GATHER_SYSTEM_STATISTICS</td>
<td>NO</td>
</tr>
<tr>
<td>GLOBAL_AQ_USR_ROLE</td>
<td>GLOBAL</td>
</tr>
<tr>
<td>HS_ADMIN_EXECUTE_ROLE</td>
<td>NO</td>
</tr>
</tbody>
</table>
Tips for Grouping Users

• Create separate groups (roles) for system and object privileges.  
  (Provides better control of privileges for the system roles and data owners 
  to grant privileges to the object roles exclusively.)

• Choose a naming convention that reflects each type of group/role (e.g. LANDBASE_EDITORS, PUBLIC_ACCESS, etc..)

• Grant privileges directly to the ArcSDE administrator user 
  and grant privileges via groups (roles) for all other users.

• Avoid mixing roles with directly granted privileges for end 
  user accounts.  
  (When end user accounts receive privileges through both roles and direct 
  grants, a well-planned security model can quickly devolve into an 
  unmanageable mess.)
Roles and OS Groups

- Support, use and configuration varies between RDBMS
- SQL Server
  - connect, read and edit data supported in 9.x and later
  - use of groups that contain members who can own data in 9.2 and later releases
  - But, the schema of the user must have the same name as the login of the individual user. You **cannot** create one schema to store the data created by all the group members.
  - permissions on the server and in individual databases is inherited from their group membership.
Thank You

http://www.esri.com/sessionevals