Database/Security Best Practices at

by
Paul P. Ruais

Presentation Overview

- philosophy and its implementation
- database/security best practices
- benefits from the practices
- presentation summary
- questions/comments

Our Philosophy + Implementation

- DBAs* and SAs** have a partnership
- DBAs and Developers work together
- enable/support rapid application development
- provide tools that support policies/procedures

* DBAs - Database Administrators
** SAs - Security Administrators

How does it really work?

I do as I’m told by Joanne

DBAs and SAs are partners

- we share information about applications
- we define database policies/procedures together
- we develop procedural tools together
- SAs are security architects/overseers and not necessarily physical implementers

DBAs and Developers are also partners

- DBAs and Developers work closely to define DB objects
- DBAs apply privileges for “programmers” to DB objects in all environments
- DBAs assist Developers in communicating their security needs to the SAs

Enable and support rapid application development

- document DB objects and their application use
- DBAs define GRANTS in all environments and the SAs review the privileges set
- SAs define all machine privileges, accounts and verify the migrated GRANT(s)

Provide tools that support our policies/procedures

- TogetherSoft's Together UML tool is used to model all DB objects and define their use
- MicroSoft's Outlook e-mail product is used to notify SAs that objects are being created as early in the development lifecycle as possible

Provide tools that support our policies/procedures (continued)

- WebSphere “will” be used to present ERDs from Together – currently we use a file share option on the DBA’s web-site
- BMC’s SmartDBA and Change Manager products are used for object migration, management and audit

Best Practices

- close “windows, doors, and holes in the wall”
- authenticate appropriately for the application
- authorize appropriately for the DB managers
- grant privileges appropriately for the users
- audit appropriately and as needed

Close the “windows”

- disable database discovery
  - do not allow the database to be discovered on the network via COMMAND line processes – don’t allow peeping Toms on your network

Close the “doors”

- eliminate database defaults and protect DB authorities
  - everyone knows the vendor’s default account name
  - restrict SYSADM, SYSCTRL, SYSMAINT, and DBADM authority levels to a very limited number of staff members

Close the “holes”

- eliminate database defaults and protect DB authorities
  - do not use the vendor’s default database administration server name nor their default instance name
  - do not create vendor’s default database in production
  - do not use database vendor’s published sample naming convention

Close the “holes” (continued)

- eliminate PUBLIC accesses to the CATALOG
  - SYSCAT.DBAUTH
  - SYSCAT.TABAUTH
  - SYSCAT.PACKAGEAUTH
  - SYSCAT.INDEXAUTH
  - SYSCAT.COLAUTH
  - SYSCAT.PASSTHRUAUTH
  - SYSCAT.SCHEMAAUTH

authenticate appropriately

- authenticate at the appropriate levels for the application and data sensitivity
  - RSA SecurID front ended as necessary
  - server or server encryption
  - client (trusted vs. untrusted)
  - DCE client/server or DCE client/server encryption
  - KERBEROS or KERBEROS server encryption

authorize appropriately

- authorize users according to database and data management needs
  - restrict SYSADM to DBAs with DBMS environment management responsibility
  - restrict SYSCTRL and SYSMAINT to staff with operational and some management responsibility

authorize appropriately (continued)

- authorize users according to database and data management needs
  - restrict DBADM authority when DB specific object privileges cannot meet your needs
  - restrict LOAD authority to data managers only when privileges cannot meet your table management needs

grant privileges appropriately

- ***EXPLICITLY*** GRANT privileges
  - different accounts are used for select vs. update
  - server pass thru and database connection
  - packages and dynamic SQL
  - schema and table
  - index and columns
  - all application objects
  - restrict PUBLIC access wherever possible

grant CONTROL or GRANT OPTION appropriately

- ***DO NOT*** GRANT CONTROL or GRANT OPTION privileges to users
  - granting CONTROL or GRANT OPTION allows the grantee to DROP the object
  - granting CONTROL or GRANT OPTION allows the grantee to GRANT other users privileges on the object
  - granting CONTROL or GRANT OPTION also implies that the grantee has REVOKE privileges on the object

audit at the appropriate levels

- DBAs control all object creation and maintenance across all environments
  - DDL/DML is managed at the server, database and schema levels using BMC's Change Manager
  - all databases have logging which allows us to monitor: who, what, when, where and how
  - ultimately, SAs have the final responsibility and authority for permissions and privileges

Benefits

- improved communication among the groups
- improved workflow streamlines the process
- increased productivity across all groups
- meta data is more complete, accurate, timely and useful
- all groups participate fully in their areas of responsibility
- objects are secured at appropriate levels

Summary

- project team is empowered and skills sets are leveraged
- controls are built into the process
- procedures are followed more consistently
- applications are delivered more rapidly
- turf wars are reduced as a team concept is employed
- resources are protected appropriately

Questions?

Comments!

Thank You.

GO TERRIERS!