Secure Schemata:
What No One is Telling You

IOUG Live! 2003 – Paper 549

Barry Johnson

~~BJohnson@WorldBank.Org~~

info@datademythed.com

~~World Bank~~

~~Washington DC~~

“Conventional Wisdom”

- DML Triggers
- Password-protected Roles
- Password Management
- Virtual Private Data Bases
- Audit Trail
- DBMS_Obfuscation
- Advanced Security Option
  - “… additional security features…” !?

What’s Wrong With “Conventional Wisdom”?

- “Synergistic wrappers” are only “speed bumps”!

    Security must be part of the basic design

- Ignores changes to Oracle’s architecture
- Ignores security requirements

    No shared authentication – ***DBAs included***

    Capture “fingerprints” of who does what

- Assumes security can be ***added***

    It can’t!

Schema: Definition

"A *schema* is a security domain that can contain database objects … *unique schemas* [sic] do not allow connections to the database …"

Oracle9*i* Application Developer's Guide – Fundamentals (9.0.1)
Ch. 11: Database Security Overview for Application Developers

User vs. Schema

- User

  Maybe defined outside the data base

  Single, maybe-external authentication

- Schema

  Owns objects

  No CREATE SESSION privilege

  "Locked down" as hard as possible!

Bug: loadjava's –schema ignored by –resolve

- ***Workarounds***: connect as owning Schema; or ALTER JAVA … RESOLVE later

Resource-limitting Profile

```
CREATE PROFILE Schema__Profile LIMIT
    COMPOSITE_LIMIT             1
    CONNECT_TIME                1 -- Unit: Minutes
    CPU_PER_CALL                1 -- Unit: 0.01sec!
    CPU_PER_SESSION             1 -- Unit: 0.01sec!
    IDLE_TIME                   1 -- Unit: Minutes
    LOGICAL_READS_PER_CALL      1
    LOGICAL_READS_PER_SESSION   1
    SESSIONS_PER_USER           1
    FAILED_LOGIN_ATTEMPTS       1

    PASSWORD_GRACE_TIME         0
    PASSWORD_LIFETIME           0
    PASSWORD_LOCK_TIME     999999
    PASSWORD_REUSE_MAX     999999
    PASSWORD_REUSE_TIME UNLIMITED ;
```

Schema Creation

```
CREATE USER <schema>
    IDENTIFIED BY VALUES 'Schema: Locked'
    PROFILE Schema__Profile
    ACCOUNT LOCK
    PASSWORD EXPIRE
    DEFAULT   TABLESPACE   <schema default>
        QUOTA UNLIMITED ON <schema default>
    TEMPORARY TABLESPACE   <instance default> ;
```

Schema Administration

- Two approaches:
  - Direct: GRANT "ANY" privileges
  - Indirect: <*schema*>-owned procedures
- Probably use both
  - Use former until latter equivalents implemented?
  - Some things *require* the latter

Schema Administration: Direct Privileges

- `GRANT CREATE TABLE ON <Schema> TO ... ;`

  ... we wish! Try again ...
- `GRANT CREATE ANY TABLE TO ... ;`

  I.e., omnipotent ...
- <*Schema Admin*> ROLE to limit use ...

  ... including getting the "ANY" privileges

Limiting Privileges:
Schema Trigger - NOT

```
CREATE TRIGGER <Schema>.Schema__Role__Check
    BEFORE DDL ON <Schema>.SCHEMA
    BEGIN
    IF NOT DBMS_Session.Is_Role_Enabled(
            '<Schema Admin>' ) THEN
        ... abort surreptitiously!? ...
        END IF ;
    END ;
```

Except: fires for **Session User**, ***not*** object **<schema>**!?!

- *Tried reporting it as a bug, told it was a feature!*
- *So much for the documented "schema" definition!!*
- *Makes it a pretty pointless feature!!!*

Limiting Privileges:
Data Base Trigger!

```
CREATE TRIGGER <Schema>.Schema__Role__Check
    BEFORE DDL ON DATABASE
    WHEN( Ora_Dict_Obj_Owner = '<Schema>' )
    BEGIN
    IF NOT DBMS_Session.Is_Role_Enabled(
            '<Schema Admin>' ) THEN
        ... abort surreptitiously!? ...
        END IF ;
    END ;
```

- Maybe check for DBA and let it through too!?

Oracle9i Application Developer's Guide – Fundamentals (9.0.1)
Ch. 13: Working With System Events

Schema Administration: Indirect Privileges

```
CREATE PACKAGE <Schema>.DDL AUTHID DEFINER AS ...
GRANT EXECUTE ON <Schema>.DDL TO <Schema Admin> ;
```

- "Must Have" Entry Points:
  - GRANT and REVOKE ... *until 9iR2*
  - CREATE private DATABASE LINK
  - RENAME
- Directly GRANT System Privileges it uses:
  - e.g. `GRANT CREATE DATABASE LINK TO <Schema> ;`
- "Dummy" procedure for initial GRANT EXECUTE!

E.g., GRANT Procedure

```
PROCEDURE Give( How   IN VARCHAR2,
                What  IN VARCHAR2,
                Whom  IN VARCHAR2 ) IS
    BEGIN
    EXECUTE IMMEDIATE
        'GRANT ' || How
       || ' ON ' || What
       || ' TO ' || Whom ;
    END Give ;
```

- ***Never*** accept and execute arbitrary SQL!

  Beware "SQL injection" as well
- ***Exception Handling***: watch what you "say"!
- ***Maybe(/probably)*** add check that “admin” Role enabled

Data Base Links

- DB Link includes Username/Password?
  - If “yes”: ***always*** a private link
  - <*schema*> VIEW/SYNONYM to remote object
  - Managed access to local reference
- Local control of remote object access
  - Due diligence/custodianship

Bug: Audit doesn’t capture private link Schema

- Fixed in a post-9iR2 release!?
- ***Workaround***: you can probably infer it!?

Referencing Objects in Other Schemata

- Views of other Tables/Views
  - Especially remote objects: get local column list
- Schema-owned SYNONYMs for other objects
  - ***No*** PUBLIC SYNONYM dependencies …
    - … except ***maybe*** Oracle's standard stuff
  - ***Not*** for TYPEs ... until 9*iR*2!
    - I.e., TYPE-owning schema must be specified
- Schema "self-contained" and "predictable"
  - Local control of remote object access
  - Due diligence/custodianship

“DDL” PACKAGE

- Example Entry Points
  - GRANTs and REVOKEs
  - Private DB link management
  - Maintain VIEWs/SYNONYMs to objects in other schemata
  - Generate standard TRIGGERs, GRANTs
- Add sophistication; e.g., for private DB link:
  - Test SELECT against User_Users at other end.
  - Create local views of remote catalog objects and GRANT to <*schema admin*> ROLE.

SYS_CONTEXT('UserEnv','<of interest>')

- Session_User: login user
- Current_Schema: implied <*schema*> default
- Current_User: current security domain
  - Procedure's <*schema*> when AUTHID DEFINER
  - Views implicitly DEFINER, but special handling
  - ***Bug***: PL/SQL returns Session_User instead!
    - ***Workaround***: "SELECT ... FROM Dual" 'til > 9*i*R2!?
  - ***Bug***: Some User_~ Views use ***Session_User***!
    - Fix available; ***no*** good workaround
- Proxy_User: trusted "external" authenticator

Current_Schema

```
ALTER SESSION SET Current_Schema = <schema> ;
```

- Implied <*schema*> when none given
  - Can***not*** define a default for a user
  - Can***not*** set a default via Login TRIGGER
  - ***Bug:*** resolving private DB links in <*schema*> views
    - Fix "in the works"
- Self-contained <*schema*>
  - VIEWs/SYNONYMs to objects in other schemata
    - Not for TYPEs ... until 9*iR2*
    - ***No*** PUBLIC SYNONYM dependencies!
  - Remote *and* local objects

DML Privileges and Roles

- ***Only*** SELECT, EXECUTE GRANTed!
  - Maybe a bit more in development space …
  - … but behind a non-DEFAULT ROLE!?
- ***Schema-owned update procedures***
  - ***Not*** necessarily the ***same*** <*schema*>
    - INSERT, etc. OK from one <*schema*> to another
  - ***Safely*** called from anywhere
  - ***Single*** call for ***all*** changes for a consistent update
  - ***COMMITs*** before returning
  - Can***not*** trust ***anything*** outside data base

“Conventional Wisdom” Review

- From earlier:
  - DML Triggers
  - Password-protected Roles
  - Password Management
  - Virtual Data Bases
  - Audit Trail
  - DBMS_Obfuscation
  - Advanced Security Option
- ***None*** of them mentioned!
  - ... *but still some* "supporting role" uses

Audit

- Failures BY ACCESS
- Successes BY ACCESS except DMLs BY SESSION
  - DELETE, EXECUTE, INSERT, LOCK, SELECT, UPDATE
- Work backwards from there
  - NO AUDIT for Dual, ~$ objects … *except* Aud$!
  - NO AUDIT for Sys.STANDARD, DBMS_STANDARD
- *Also*: Log Miner

Audit Trail Tablespace

Note:
Moving the `SYS.AUD$` table out of the `SYSTEM` tablespace is not supported because the Oracle code makes implicit assumptions about the data dictionary tables, such as `SYS.AUD$`, which could cause problems with upgrades and backup/recovery scenarios.

Oracle9*i* Database Administrator's Guide (9.0.1)
Ch. 26: Auditing Database Use

More Important Features

- Advanced Security (8*i* ASO, 9*i* AS)
    - [SQL*]Net[8] and JDBC encryption
        - ***Secure application impossible without it!***
    - External Authentications: RADIUS, Kerberos, PKI
    - * ***Extra-cost option!!***
- CONTEXTs
    - Maintain Session state information
    - Its own namespace
        - I.e., may have same name as schema!
    - ***Can*** be set by Login TRIGGER

Secure Schemata: Summary

- "Locked down" Schema
  - ***No*** shared authentications
- Administration
  - "Filtered" use of "ANY" privileges
  - Schema-owned "DDL" procedures
  - Authenticated DB Links are ***always*** private
- SELECT, EXECUTE DML privileges ***only***
  - Application-specific "safe" update procedures
- Say "schema", *not* "owner"!!