Agenda

Oracle 12c Multitenant

1. Security Impact

2. Recommendations

3. Q&A
Agenda

Oracle 12c Multitenant

1. Security Impact
2. Recommendations
3. Security Impact
4. Q&A
Oracle 12c

- **Major new features**
  - In-memory*
  - Multitenant (pluggable databases)*

- **Incremental security improvements**
  - Oracle Database Vault (DV) pre-installed*
  - Data Redaction*
  - Real Application Security
  - Unified Auditing
  - Mandatory Auditing

- **Oracle 12.2 released November 2016**
  - Currently only available on Oracle Cloud
  - On-premise? Ask your Oracle Sales rep.

* Additional license option
Why Oracle Multitenant?

- Why virtualize servers if the end goal is to consolidate databases?
  - Avoid VM Sprawl and virtualize just the databases
  - No application or code changes to use

- Benefits of virtualization
  - Increase labor efficiency of DBAs to maintain
  - Realize infrastructure cost savings by increasing density and reducing physical hardware
Containers, Seeds, Roots and Plugs

- **Container database (CDB) is the host**
  - Container for guest databases
  - Configurations stored in “Root” database (CDB$ROOT)
  - Documentation and dictionary also refer to as ‘Common’
  - Metadata and common users and objects

- **All CDBs have a PDB$SEED database**
  - Used as a template to create new PDBs
  - DO NOT alter or change anything in PDB$SEED

- **Guest databases referred to as Pluggable Databases (PDBs)**
  - Each PDB is isolated ‘sandboxed’ from all other PDBs
  - Unplugged PDB consists of XML file to describe the PDB and PDB's files (e.g. data files and/or wallet)
  - With Oracle12.2 max of 4,096 PDBs per CDB
Oracle 12c Multitenant Consolidation

- Assume all servers can be and/or are virtualized
- PDBs move between CDBs
Oracle 12c Multitenant Architecture
Security is a process
  - Created by people following processes using tools

Security requires defense-in-depth
  - Failures can, have and will always occur
  - Preventive and detective controls plus response

Security requires trusting people
  - Need trust-but-verify, especially DBAs

This presentation is based on Integrigy’s research and database security Framework
  - Integrigy’s Audit methodology
  - What could go wrong and how
Listener Security

- Single SQLNET.ORA file
  - Applies to all PDBs

- Single LISTENER.ORA file
  - Each PDB is a service
  - PDB automatically added when created

- Security impact to listener
  - Does anything change?
Multitenant Container Databases

- **Use Container databases for**
  - Hosting PDB guest databases
  - Defining common user, roles and security and audit policies

- **Do not use Container database for building applications**
  - Tablespaces, Tables, Application users and roles, Directories, Database links, Public database links

- **Audit and monitor changes to CDB$ROOT**
  - Question creative ideas
New CDB_XXXX Dictionary Views

- **CDB_XXXS All objects in the CDB across all PDBs**
- **DBA_XXSX All objects in CDB or PDB**
- **ALL_XXSX All objects accessible by user**
- **USER_XSXX All objects owned by user**

Security audits need to incorporate new CDB_XXX objects.
Pay close attention to *oracle_maintained* = ‘N’ or ‘Y’
Startup Parameters

- **All PDB inherit startup parameters from CDB**
  - Subset can be overridden with PDB
  - Overrides stored in PDB_SPFILE$

- **ISPDB_MODIFIABLE determines if PDB can change**
  - Cannot be changed (199): Auditing, FIPS-140, UTL_FILE_DIR
  - Can be changed (184): NLS, O7_DICT, Sessions
    
    \[ v\$system\_parameter \text{ WHERE } ispdb\_modifiable = 'TRUE' \]

- **Audit and monitor parameter changes for both CDB and PDB**
  - Add to your monitoring and audit scripts
Multitenant Patching

- **Patches ONLY applied to CDB**
  - Container and all PDBs all exact same version
  - Patches cannot be applied to PDBs

- **How test patches?**
  - Unplug and plug into CDB @ different patch level
Multitenant Is Great For CPU Patching

- Only CDB is patched
- CPU applied once to CDB
Two Types of Users

- **Common user**
  - Exists in ALL current and *future* PDBs
  - Oracle default accounts are common: SYS, SYSTEM, CTXSYS, ...
  - O/S authentication is allowed (not recommended)
  - External authentication allowed (not Oracle recommended)
  - NOT moved when PDB plugged into another CDB
  - Username must use prefix (default is C##)
  - Identified in CDB_USERS where common = ‘YES’

- **Local user**
  - A user local to a single PDB
  - Username CANNOT use C## prefix
  - O/S authenticated NOT allowed
  - Can use external authentication like SSL, Kerberos
  - Can have SYSDBA rights local to PDB
Multitenant User Security Recommendations

- **Create common users sparingly**
  - Cost of complexity

- **Can restrict common user access to specific PDBs**
  - Remove ‘create session’ to specific PDBs
  - Use ‘container data’ to whitelist PDBs when creating users

- **Audit and monitor creation and status of common users**
  - With 12c all default Oracle accounts except SYS and SYSTEM are expired and locked
  - In DBA_USERS use new ORACLE_MAINTAINED and COMMON columns to differentiate Oracle created and common users
  - Common users should not own local PDB objects
Can You Issue DML to Other PDBs?

- Common users can access (switch to) ANY current and future PDB
  - Can be whitelisted and restricted

- **12.2 CDB users can query PDBs with CONTAINERS clause**
  - Objects must be owned by common user issuing SQL
    ```sql
    SELECT * FROM CONTAINERS(employees) WHERE CON_ID IN(3,4);
    ```

- **PDBs cannot query other PDBs**
  - SGA is logically virtualized
  - 
    ```sql
    EXECUTE IMMEDIATE ALTER SESSION SET CONTAINER
    ```
    to set another container is blocked within PL/SQL

- **Database links between PDBs are allowed**
  - As are between non-multitenant databases and PDBs
Two Types of Roles

- **Common role**
  - Exists in ALL current and *future* PDBs
  - All Default roles are common
  - Role name **must** use prefix (C## default)
  - Not moved when PDB plugged into another CDB
  - Common roles granted to local users have role only with in local PDB

- **Local role**
  - Role name **cannot** use C## prefix
Multitenant Role Security Recommendations

- **Create and use common role sparingly**
  - Cost of complexity

- **Audit and monitor creation and status of common roles**
  - In DBA.Roles use new `ORACLE_MAINTAINED` and `COMMON` to differentiate

- **Grant SET CONTAINER privilege with caution**
  - Allows user to connect to any PDB within CDB without authenticating
New Flavors of DBAs

- CDB_DBA Role
  - Common role for container administration

- PDB_DBA Role
  - Local role exists only in PDB for administrative tasks

- PDB User PDBADMIN
  - Created by default within each PDB
  - By default gets PDB_DBA role

- Recommend to monitor, audit and alert
  - Explain to auditors and IT security
Two Types of PUBLIC GRANTS

- **Common PUBLIC grants**
  - Granted in CDB and given to PUBLIC in all PDBs
  - Cannot be altered within PDB

- **PDB PUBLIC grants**
  - Local to the PDB

- **Recommendations**
  - Avoid common PUBLIC grants
  - Audit and monitor for abuse
Triggers With Oracle 12c Multitenant

- Database event triggers can be created in CDBs or PDBs
  - New events added for managing and moving among PDBs

- Logon and DML triggers often used for auditing. If using, be sure to consider:
  - AFTER LOGON
  - BEFORE LOGOFF
  - BEFORE SET CONTAINER
  - AFTER SET CONTAINER
Two Types of Profiles

- **Common profiles**
  - Exists in ALL current and future PDBs
  - Not moved when PDB plugged into another CDB
  - Use C## prefix

- **Local profile**
  - Same as before

- **Audit and monitor profile changes for both CDB and PDB**
  - Add to your monitoring and audit scripts
12.2 Multitenant PDB Lockdown Profiles

- New with 12.2 can restrict features and options available in PDBs
  - Different than resource limit profile
  - Assign to individual PDBs, or to all PDBs in a CDB

- Examples
  - O/S & Network access
  - System privileges

- 12.1 requires manually restricting privileges grants, and configurations in each PDB
12.1 Multitenant File System Access

- One (1) Oracle_Home can support more than CDB
  - Shared by ALL current and future PDBs
  - Owned by single O/S account e.g. ‘Oracle’

- What about PDB file system access?
  - UTL_FILE and Directories
  - External tables & SQL-LOADER
  - EXTPROC

- Need to manually isolate each PDB within each CDB
  - DB and O/S grants and configurations
  - PDB DBAs, Applications and developers
  - Audit and monitor for abuse
  - Fixed in 12.2
12.2 Multitenant File System Access

- **12.2 new startup parameters**
  - PDB_OS_CREDENTIAL – dedicated O/S user for a PDB
  - PATH_PREFIX and CREATE_FILE_DEST – isolates PDB files to a specified directory and its subdirectories

- **UTIL_FILE**
  - Use Oracle Directories instead within PATH_PREFIX
  - 12.2 deprecates UTL_FILE_DIR. Supported but Oracle recommends not to use.

- **External tables**
  - Define using path within PDB’s PATH_PREFIX AND CREATE_FILE_DEST’

- **EXTPROC**
  - Use account specified in PDB_OS_CREDENTIAL

- **12.2 only available on Oracle Cloud**
  - Ask Oracle sales rep about on-premise
Oracle 12c Rewrite of Auditing

- **Unified Auditing**
  - New schemas, features, queuing modes, syntax
  - Two (2) modes

- **Pure Mode Unified Auditing**
  - Only 12c Unified Audit functionality available
  - No Syslog

- **Mixed Mode (Default) Unified Auditing**
  - Has both **Traditional** and Unified Auditing
  - Provided as introduction and transition
Oracle 12c Multitenant Auditing

- **One Alert log for CDB and ALL PDBs**
  - `<diagnostic_dest>/diag/rdbms/<CDB NAME>/<CDB INSTANCE>/trace`

- **Traditional or Unified Auditing**
  - Same for CDB and ALL PDBs

- **Each PDB and the CDB has own audit trail**
  - Each has SYSAUX tablespace and UNIFIED_AUDIT_TRAIL
  - `CDB_UNIFIED_AUDIT_TRAIL` has ALL PDB audit activity

- **Common vs. Local Audit policies**
  - Common audit policies for common objects only and local audit policies for local objects only
  - Common policies **NOT** moved when PDB moves
Other Oracle 12c Multitenant Security

- VPD policies local to PDB only
- **Transparent Sensitive Data Protection (TSPD) local to PDB**
  - New with Oracle 12c
- **Transport Layer Security (SSL)**
  - Each PDB must have own wallet and SSL certs
Transparent Database Encryption (TDE)

- **TDE only encrypts “data at rest”**
  - Requires no application code or database structure changes
  - Additional license option to use

- **TDE provides coarse-grained security by controlling access to data files**
  - Once data is in-memory it is **NOT** encrypted
  - Protects storage media (disk or tape) if stolen, lost or hacked

- **TDE is supported by Multitenant**
  - Keystore (Wallet) exists in O/S not in PDB(s)
  - Each PDB has own TDE master encryption key
Oracle Data Vault

- **Installed by Default with Oracle 12c**
  - Requires no application code or database structure changes to implement
  - Additional license option to use

- **Data Vault provides medium-grained security**
  - Secures SYS and SYSTEM users
  - Can “blind” DBAs from seeing sensitive data e.g. cannot use SELECT with EBS ‘APPS’ schema
  - Use FGA, VPD or ASO for fine-grained security

- **Data Vault is supported by Multitenant**
  - ALL PDBs do **NOT** ALL need to use Data Vault
  - Can apply to just CDB? We are researching this.
  - Do not allow DV_OWNER account to be locked
Agenda

1. Oracle 12c Multitenant
2. Security Impact
3. Recommendations
4. Q&A
Before Start Using Multitenant

- **Vet Applications**
  - Dictionary access & UTL_FILE

- **Vet compliance requirements**
  - PCI, SOX, DISA STIG

- **Have solution to secure unplugged PDBs**
  - Just like VM guest images

- **Revise DBA policies and assignments**
  - CDB_DBA, PDB_DBA PDBADMIN
  - Explain to auditors and IT security

- **Revise database and IT security policy**
  - No trampoline policy!

- **Update all audit and monitoring scripts**
  - New events, objects, parameters and privileges
Before Start Using Multitenant

- **Segregate CBDs by production status**
  - Use separate CDBs for product/non-production

- **Only customize PDBs**
  - Change CDB sparingly
  - Do not customize PDB$SEED

- **Do not change common user prefix C##**
  - Startup parameter `common_user_prefix`

- **Remove APEX from CDB**
  - Install in PDB as needed

- **Convert now to Oracle Directories**
  - Stop using `UTIL_FILE`

- **Upgrade to 12.2 when available**
  - Ask your Oracle sales rep

- **Use PDB Lockdown profiles (once on 12.2)**
  - Pay close attention to O/S restrictions
Common Users and Roles

- **Create for maintenance not Application purposes**
  - Minimally privilege with standardized across CDB, CDBs and all PDBs
  - Don’t authenticate externally
  - Avoid complexity

- **Be careful about**
  - Granting *CREATE SESSION* commonly as it gives access to all current and future PDBs
  - Granting *SET CONTAINER* commonly as it allows users to move among PDBs without authenticating
  - Keeping default Oracle accounts locked and expired
Use Multitenant To Strengthen Security

- **Use Oracle 12c Multitenant to implement or strengthen database security program**
  - Oracle 12c Multitenant consolidates and standardizes databases

- **Use Integrigy database security framework**
  - Approach defines a common framework for all databases
  - See [www.integrigy.com](http://www.integrigy.com) for more information
#1 Recommendation is to reduce security vulnerability exposure

- Use both virtual and physical perimeters to reduce access to databases
- Standardized secure configuration baseline
Framework = Consistency

Oracle 12c Multitenant makes this easier
Oracle 12c Multitenant Allows For

- **Consistent virtualized perimeters**
  - Consolidated servers and PDBs

- **Consistent patch levels**
  - Only CDB is patched

- **Consistent inherited security best practices**
  - Startup parameters
  - Users
  - Roles
  - Profiles
  - Lockdown profiles
  - Audit policies
Database Security Program Silos

Processes should be unified, but standards and procedures need to be vendor specific.
# Database Security Program Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Inventory** | - An inventory of all databases and sensitive data locations  
                          - Methods and processes to maintain the inventories |
| **Configuration** | - A measurable database security standard and baseline  
                          - Periodic validation with compliance to the standard |
| **Access**     | - Database access management policies, procedures, and tools  
                          - Database access profiling and monitoring |
| **Auditing**   | - Database auditing requirements, processes, and definitions  
                          - Centralized auditing retention and reporting solution |
| **Monitoring** | - Database real-time security monitoring and intrusion detection  
                          - Database monitoring definition and tools |
| **Vulnerability** | - Vulnerability assessment and management for databases  
                          - Vulnerability remediation strategy and processes |
| **Protection** | - Sensitive data protection strategy – encryption, data masking, redaction, scrambling  
                          - Data protection policies, procedures, and tools |
Contact Information

Michael Miller
Chief Security Officer

web: www.integrigy.com
e-mail: info@integrigy.com
blog: integrigy.com/oracle-security-blog
youtube: youtube.com/integrigy
Program Implementation

Inventory
- DB Discovery
- Data Discovery
- Update Change Mgmt
- Living DB Inventory
- Living Data Inventory

Configuration
- Configuration Standards
- DB Access Management Definition
- Implement Access Solution
- Access Profiling
- Baseline Database Auditing
- Key Application Auditing
- Database IDS
- Log Monitoring Integration
- Implement Configuration Std
- Periodic Vulnerability Scans

Access
- DB Access Management Definition
- Access Controls/Policies
- Access Profiling

Auditing
- Configuration Standards
- Configuration Standard Auditing

Monitoring
- DAM Definition and Architecture
- DAM Selection and Implement
- Solution Selection and Implementation
- Data Protection Process

Vulnerability
- Implementation Requirements
- Periodic Vulnerability Scans

Protection
- On-going