Oracle Database 12c Enterprise Edition

Guidance Supplement

Evaluation Assurance Level (EAL): EAL2+

Doc No: 1932-000-D105
Version: 1.2
6 March 2017

Oracle Corporation
5000 Oracle Parkway
Redwood Shores, California
94065

Prepared by:
EWA-Canada
1223 Michael Street, Suite 200
Ottawa, Ontario, Canada
K1J7T2

CONTENTS

1 SECURE ACCEPTANCE PROCEDURES

Secure acceptance procedures ensure that the correct version of the TOE has been received by the customer as intended by the developer. Oracle Database 12c may be downloaded by registered users from the Oracle secure delivery cloud at https://edelivery.oracle.com/.

After accepting the license agreement and the export restrictions, the user may then select the product pack (Oracle Database) and the platform (Linux x86-64), and select 'Go'. The user may then select the product (Oracle Database 12c Release 1 (12.1.0.2.0) Media Pack for Linux x86-64) and select 'Continue'. A list of files appears. A 'Readme' button opens a window with further instructions for the download. A 'View Digest' button opens a window with MD5 and SHA-1 digests for each of the zipped files. The files may be downloaded by selecting the 'Download' button. The user may then use a third-party application to verify the digest before proceeding to unzip and install the files.

1.1 PATCH AND CRITICAL UPDATES (PPU/CSU)

Information on the January 2017 Patch/Critical Patch Update can be found at:

http://www.oracle.com/technetwork/security-advisory/cpujan2017-2881727.html

1. To download the patch a user needs to access the Oracle support website: https://support.oracle.com.
2. Click "*Sign In*".

   Note: First time users must first register by clicking "*New User? Register here*".
3. Select the Patches and Updates tab.
4. Search by Patch Number/name: **24917069**
5. Click Search.

Patch 24917069: Combo OJVM PSU 12.1.0.2.170117 and Database PSU 12.1.0.2.170117

6. Select the patch and click on the Readme button to access instructions. Follow the Readme instructions.
7. Click Download to download the patch.
8. Click on p24917069_121020_Linux-x86-64.zip.

Additional information about the patch can be found in My Oracle Support at:

https://www.oracle.com/technetwork/topics/security/cpujan2017-2881727.html

2 SECURE INSTALLATION PROCEDURES

This section describes the steps necessary for secure installation of the TOE and the secure preparation of the operation environment in the evaluated configuration.

2.1 SECURE PREPARATION OF THE OPERATIONAL ENVIRONMENT

The following assumptions are made with respect to the secure installation of the TOE and its operational environment:

Assumptions	Description
Physical aspects	
A.PHYSICAL	It is assumed that the IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.
Personnel aspects	
A.AUTHUSER	Authorized users possess the necessary authorization to access at least some of the information managed by the TOE.
A.MANAGE	The TOE security functionality is managed by one or more competent administrators. The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the guidance documentation.
A.TRAINEDUSER	Users are sufficiently trained and trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their user data.
Procedural aspects	
A.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration, and support of the DBMS.
A.PEER_FUNC_&_MGT	All remote trusted IT systems trusted by the TSF to provide TSF data or services to the TOE, or to support the TSF in the enforcement of security policy decisions are assumed to correctly implement the functionality used by the TSF consistent with the assumptions defined for this functionality and to be properly managed and operate under security policy constraints compatible with those of the TOE.
A.SUPPORT	Any information provided by a trusted entity in the IT

Assumptions	Description
	environment and used to support the provision of time and date, information used in audit capture, user authentication, and authorization that is used by the TOE is correct and up to date.
Connectivity aspects	
A.CONNECT	All connections to and from remote trusted IT systems and between separate parts of the TSF are physically or logically protected within the TOE environment to ensure the integrity and confidentiality of the data transmitted and to ensure the authenticity of the communication end points.

Table 1 – Assumptions

The following subsections provide additional guidance required to meet the secure preparation of the operational environment.

2.1.1 OE.ADMIN

OE.ADMIN Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains.

Users of the Oracle DB12 database must ensure that only known, competent, trusted employees are made responsible for managing the security of the database and the data contained therein. Employees should be subject to background checks and undergo Oracle DB12 database training before being put into a position of trust.

2.1.2 OE.INFO_PROTECT

OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular:

- All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques.
- DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly.
- Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.

Adherence to ISO/IEC 11801 standards is required for the implementation of cabling associated with any device connected to the network which includes an Oracle DB12 database implementation. Both copper and fibre optic cabling are permitted.

Users of the Oracle DB12 database must ensure that all implementations are fully planned prior to system installation and configuration. All access controls must be put in place before the database is populated.

The Oracle DB12 database must be implemented using a 'least privilege' approach. Users may only be permitted access to the data to which access is required in order to perform assigned functions. Only those users fully trained in the use of the Oracle DB12 database, and who have been advised of their privileges and responsibilities may be given access.

2.1.3 OE.NO_GENERAL_PURPOSE

OE.NO_GENERAL_PURPOSE	There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration, and support of the DBMS.

Installers of the database must ensure a fresh installation of the underlying operating system has been implemented and hardened in accordance with the organization's best practices prior to database installation. Access to the operating system must be strictly controlled, and no other services may be installed on the database server.

2.1.4 OE.PHYSICAL

OE.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to enforcement of the security policy are protected from physical attack that might compromise IT security objectives. The protection must be commensurate with the value of the IT assets protected by the TOE.

Installers are instructed to only install the Oracle DB12 database in locations that provide physical security against possible attack in accordance with the organization's policy. Security should be increased in accordance with the value of the data to be protected within the database.

2.1.5 OE.IT_I&A

OE.IT_I&A	Any information provided by a trusted entity in the environment and used to support user authentication and authorization used by the TOE is correct and up to date.

Prior to configuring an Oracle DB12 database with an external authentication mechanism, the implementers must ensure that every entry in the authentication system is correct and up to date.

2.1.6 OE.IT_REMOTE

OE.IT_REMOTE	If the TOE relies on remote trusted IT systems to support the enforcement of its policy, those systems provide that the functions and any data used by the TOE in making policy decisions required by the TOE are sufficiently protected from any attack that may cause those functions

to provide false results.

The implementers of the Oracle DB12 database must ensure that any system that connects to the database and provides input to the database's security policy decision making must be implemented securely and protected from possible physical attack.

2.1.7 OE.IT_TRUSTED_SYSTEM

OE.IT_TRUSTED_SYSTEM	The remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy. These remote trusted IT systems are managed according to known, accepted, and trusted policies based on the same rules and policies applicable to the TOE, and are physically and logically protected equivalent to the TOE.

The Oracle DB12 database implementation team must ensure that any system that connects to the database must be implemented securely and protected from possible physical attack. Only remote systems that are under control of those implementing the database, and subject to the same physical and access control security policies should be allowed to access the database.

2.2 INITIAL SETUP AND CONFIGURATION

Administrators should perform the initial setup and configuration of the TOE in accordance with the instructions provided in the following chapters from the *Oracle® Database Installation Guide 12c Release 1 (12.1) for Linux*:

- Chapter 4, Oracle Database Preinstallation Tasks
- Chapter 5, Configuring Users, Groups and Environments for Oracle Database
- Chapter 7, Installing Oracle Database
- Chapter 8, Oracle Database Postinstallation Tasks

2.3 PASSWORD CONFIGURATION

Administrators are required to manually enable the password complexity checking function using the `Ora12c_strong_verify_function`. Instructions on enabling this function can be found in the *Oracle® Database Security Guide 12c Release 1 (12.1)* under Chapter 3, Configuring Authentication → Enabling Password Complexity Verification (page 3-17).

3 OTHER PROCEDURES

This section describes the user-accessible functions and privileges that should be controlled in a secure processing environment, and includes the security-critical information and security-critical actions required for secure use of the TOE.

3.1 INITIALIZATION PARAMETERS

The following steps must be completed for the TOE to operate in the evaluated configuration.

a) To connect to the DBMS as a privileged user, such as a database administrator, the following parameters shall be set in the appropriate initialization file:

```
O7_dictionary_accessibility = FALSE;
Remote_login_passwordfile = EXCLUSIVE
```

b) The following parameter ensures that a user must have SELECT privilege on a table when executing an UPDATE or DELETE statement that references table column values in a WHERE or SET clause:

```
sql92_security = TRUE
```

c) The `audit_trail` parameter in the appropriate initialization parameter file shall be assigned in the following ways:

```
audit_trail = DB
```

d) The following parameter enables session auditing:

```
audit session
```

e) The following parameters revoke default PUBLIC privileges:

```
revoke execute on DBMS_JOB from Public;
revoke execute on DBMS_JAVA from public;
revoke execute on DBMS_XMLGEN from public;
revoke execute on utl_smtp from public;
revoke execute on utl_tcp from public;
revoke execute on utl_http from public;
revoke execute on utl_file from public;
revoke execute on dbms_random from public;
revoke execute on SYS.OWA_OPT_LOCK from public;
revoke execute on XDB.DBMS_XDB from public;
revoke execute on CTXSYS.DRILOAD from public;
revoke execute on MDSYS.PRVT_IDX from public;
revoke execute on SYS.DBMS_CDC_DPUTIL from public;
```

```
revoke execute on SYS.DBMS_EXPORT_EXTENSION from public;
revoke execute on SYS.DBMS_TRANSFORM_EXIMP from public;
revoke execute on XDB.XDB_PITRIG_PKG from public;
revoke insert on mdsys.user_sdo_geom_metadata from public;
revoke insert on mdsys.user_sdo_lrs_metadata from public
```

f) In the evaluated configuration, the operating system does not authenticate remote users nor perform role associations. Therefore, the following parameters must be set:

```
remote_os_authent = FALSE
os_roles = FALSE
remote_os_roles = FALSE
```

g) The following parameter ensures that modifications to the roles of a user are audited:

```
audit system grant whenever not successful;
audit grant on <object> whenever not successful;
audit role whenever not successful;
```

3.2 LOGON TRIGGER CONFIGURATION

Oracle has provided example logon triggers as .sql programs available online at http://www.oracle.com/technetwork/topics/security/oracle-common-criteria-evaluated-083264.html#Frame1. From the link, select 'FTA Adapted Files' and download the zipped folder. These programs allow you to establish a trigger (or hook) that is executed after logon. The triggers use a database table where the logical expression is stored. The content of that table is then evaluated to allow or deny the logon.

Note: It is important to note that the triggers are executed after logon, and therefore you must be careful when applying logon triggers because as the rules become complex you can inadvertently lock out users (for example, user SYS).

The available .sql programs are:

- install.sql – installs logon triggers
- deinstall.sql – removes logon triggers
- audit_trail.sql – enables you to view relevant audit logs
- package.sql – the package required by install.sql

These programs can be executed by an administrator who has the privileges necessary to install triggers.

Note: The contents of each .sql package are reproduced in the corresponding subsections below.

3.2.1 Restricting Session Establishment by Time of Day and Day of Week

To restrict session establishment by time of day and day of the week, Oracle provides a set of packages that can be used to implement this requirement via an "after-logon" trigger. This section explains how to install/deinstall this function, and how to use it.

3.2.1.1 Installation

To install the trigger, the install.sql package is used. Before executed, modify the install.sql and deinstall.sql scripts according to your needs:

- id of the new user (’tsf’ by default)
- default tablespace (’sysaux’ by default)
- passwords of the new user (placeholder ’<tsfpass>’)

The after-logon trigger is then executed with the privileges of this new user. When created, the new user gets the following privileges assigned:

- Create session
- Create procedure
- Create table
  Administer database trigger
- Create trigger
- Select, insert and update on table sys.aud$
- Select, insert and update on table system.aud$
- Create role

A new role 'SECURITY_ADMIN' is created and assigned to the new user.

Note: Execution of this install package will replace any existing after-logon triggers as well as any existing before-ddl triggers. Also note that the created tsf account is normally not used afterwards, so that you might decide to expire the tsf account:

- alter user tsf account lock password expire;

3.2.1.2 Usage

Once installed, the package allows the definition of a rule used by the after-logon trigger to determine if session establishment is allowed by day of the week or time of the day. The management functions available for a user with the role of SECURITY_ADMIN are:

- add_event_rule (event, rule_expression)
  which allows adding a new rule
- update_event_rule (event, rule_expression)
  which allows updating an existing rule
- delete_event_rule (event)
  which allows deleting an existing rule

ADD_EVENT_RULE Procedure

This procedure adds an event rule that is associated with a specific event. The rule is evaluated by a trigger function associated with the event.

Syntax

```
ADMIN.ADD_EVENT_RULE (
    event           IN VARCHAR2,
    rule_expression IN VARCHAR2);
```

Parameters:

event	a string of maximum 100 characters that specifies the event. **Note**: this parameter must be set to the value 'LOGIN' to define a rule that is evaluated by the after-logon trigger.
rule_expression	a string of less than 3900 character that defines the rule to be evaluated. See the section below for details on how to define rules.

MODIFY_EVENT_RULE Procedure

This procedure adds an event rule that is associated with a specific event. The rule is evaluated by a trigger function associated with the event.

Syntax

```
ADMIN.MODIFY_EVENT_RULE (
    event           IN VARCHAR2,
    rule_expression IN VARCHAR2);
```

Parameters:

event	a string of maximum 100 characters that specifies the event. **Note**: this parameter must be set to the value LOGIN' to define a rule that is evaluated by the after-logon trigger.
rule_expression	a string of less than 3900 character that defines the rule to be evaluated. See the section below for details on how to define rules.

DELETE_EVENT_RULE Procedure

This procedure adds an event rule that is associated with a specific event. The rule is evaluated by a trigger function associated with the event.

Syntax

```
ADMIN.DELETE_EVENT_RULE (
    event           IN VARCHAR2);
```

Parameters:

event	a string of maximum 100 characters that specifies the event. **Note**: this parameter must be set to the value 'LOGIN' to define a rule that is evaluated by the after-logon trigger.

3.2.1.3 How to Define an Event Rule

To restrict login by day of the week, the rule must be constructed in the following way:

USER IN ({List of users that are not restricted by the rule})

OR

(RTRIM(TO_CHAR(SYSDATE, ' 'DAY' ')) IN ({List of days of the week a user is allowed to login}) AND
RTRIM(TO_CHAR(LOCALTIMESTAMP, ' 'HH24' ')) IN ({List of hours a user is allowed to login}))

[**Note: The entire event rule must be enclosed in single quotation marks (e.g. 'Event Rule'). Also note the use of two single quotation marks, rather than a double quotation mark. The day and hours must also be enclosed in two single quotation marks (e.g. ' ' Monday, Tuesday ' ').**]

{List of users that are not restricted by the rule}:
A comma separated list of strings where each string is the ID of a user that is not restricted by the rule.

Example: 'SYS' , 'TSF'
In this example the users SYS and TSF would not be restricted by the rule

{List of days of the week a user is allowed to login}:
A comma separated list of days of the week where users are allowed to login

Example: 'MONDAY' , 'TUESDAY , 'WEDNESDAY' , 'THURSDAY' , 'FRIDAY'
This example list would not allow users (other than the ones in the list above) to login on Saturdays and Sundays.

{List of hours a user is allowed to login}:
A comma separated list of hours of the day where users are allowed to login.

Example: '08', '09', '10', '11', '12', '13', '14', '15', '16', '17'
This example list would allow users to login at a time where the hour value at the time of login (in 24 hour format) is between 8 and 17 (i.e. between 8:00am and 5:59pm).

3.2.2 Obtaining Login Information

To obtain information about the time and location of the last successful login and the number of unsuccessful login attempts since the last successful login, a user

can call the procedure tsf_logon_status. Before calling this procedure, the user must enter the following command:

SET SERVEROUTPUT ON

[**Note: Traditional auditing, rather than Unified Auditing, must be used to support this function.**]

This provides the information in the form:

Welcome <userid>. Your last logon was on "DD-MMM-YY HH.MM.SS.mmmm AM/PM +HH:MM";

from host "<host-id>" on terminal "<terminal-id>".

There have been <n> unsuccessful logon attempts since your last logon.

Where <userid> is the name of the user calling the procedure, <host-id> is the Client host machine name where the last successful login was performed from and <terminal-id> is the identifier of the terminal used for the last successful login.

The number of unsuccessful login attempts is counted since the last successful login. This allows a user to check if someone has used his account or has unsuccessfully attempted to logon using his account. The ability to use this function requires the successful installation of the logon trigger and enabling session auditing.

3.2.3 Install Objects and Packages for Rules-Based Login Control (install.sql)

```
set echo on

connect / as sysdba

create user tsf identified by <tsfpass>
default tablespace sysaux;
alter user tsf quota 5m on sysaux;

grant create session to tsf;
grant create procedure to tsf;
grant create table to tsf;
grant administer database trigger to tsf;
grant create trigger to tsf;
grant select , insert, update on sys.aud$ to tsf ;

-- if OLS installed

grant select, insert, update on system.aud$ to tsf ;
grant select, insert, update on sys.dba_audit_trail to tsf ;
grant select, insert, update on sys.dba_audit_session to tsf ;
grant create role to tsf;

connect tsf/<tsfpass>;

create role security_admin;
```

```
-- Note: you could extend the concept of the event rule to have audit codes
-- and audit messages so the auditing is more customizable

        create table security_criteria (
        event_name varchar2(100) not null
        , event_rule varchar2(4000) not null
        );

alter table security_criteria add constraint security_criteria_pk
primary key (event_name) enable;

@@package.sql

create or replace trigger after_logon_trigger after logon on database
begin

        admin.evaluate_rule('LOGIN');
exception

        when others then
            raise;
end;
/

connect / as sysdba
grant execute on tsf.logon_status to public;
grant execute on tsf.logon_last_host to public;
grant execute on tsf.logon_last_terminal to public;
grant execute on tsf.logon_last_date to public;
grant execute on tsf.logon_unsuccessful_count to public;
create or replace public synonym tsf_logon_status for tsf.logon_status;
create or replace public synonym tsf_logon_last_host for tsf.logon_last_host;
create or replace public synonym tsf_logon_last_terminal for
tsf.logon_last_terminal;
create or replace public synonym tsf_logon_last_date for tsf.logon_last_date;
create or replace public synonym tsf_logon_unsuccessful_count for
tsf.logon_unsuccessful_count;
alter system set audit_trail = db scope = spfile;
alter system set audit_sys_operations = true scope = spfile;

AUDIT CREATE SESSION BY ACCESS WHENEVER SUCCESSFUL;
AUDIT CREATE SESSION BY ACCESS WHENEVER NOT SUCCESSFUL;
```

3.2.4 De-Install Rules-Based Login Control (deinstall.sql)

Note: The following deinstalls only the default user that was assigned during install (see "install.sql"). If you defined a user other than the default, you must modify the script below to deinstall the defined user.

```
connect / as sysdba
drop public synonym tsf_logon_status;
drop public synonym tsf_logon_last_host;
```

```
drop public synonym tsf_logon_last_terminal;
drop public synonym tsf_logon_last_date;
drop public synonym tsf_logon_unsuccessful_count;
drop user tsf cascade;
drop user tsfuser cascade;
drop role security_admin;
```

3.2.5 Display for Audit Trail of Rules-Based Login Control (audit_trail.sql)

```
column action_name format a15
column username format a15
column comment_text format a30
select to_char(cast(extended_timestamp as date),'DD-MON-YYYY HH24:MI:SS')
    , username,action,action_name,returncode
from dba_audit_trail
where extended_timestamp > (sysdate-1)
    and action_name like 'LOG%'
order by extended_timestamp
/
```

3.2.6 Package Rules-Based Login Control (package.sql)

```
set echo on
        create or replace package admin as
        procedure add_event_rule ( event varchar2, rule_expression varchar2 );
        procedure update_event_rule ( event varchar2, rule_expression varchar2
        );
        procedure delete_event_rule ( event varchar2 );
        procedure evaluate_rule ( event varchar2 );
        procedure logon_status;
        function logon_last_date return varchar2;
        function logon_last_host return varchar2;
        function logon_last_terminal return varchar2;
        function logon_unsuccessful_count return number;
        function session_has_role ( role_name varchar2 ) return number ;

end;

/
```

3.3 NETWORK ENCRYPTION CONFIGURATION

Network encryption is outside the scope of the evaluation. However, an administrator can manually enable the encryption of data that is sent over the network. Administrators should configure the network encryption in accordance with the instructions provided in the following chapters of the *Oracle® Database Security Guide 12c Release 1 (12.1)*:

- Chapter 13, Configuring Data Encryption and Integrity
- Chapter 14, Configuring the Thin JDBC Client Network

4 APPENDIX A – REFERENCES

The following installation and administrative guides are referenced within this document:

- Oracle® Database Installation Guide 12c Release 1 (12.1) for Linux, Part Number E41491-16
- Oracle® Database Security Guide 12c Release 1 (12.1), Part Number E48135-15