

# ORACLE 11G DATABASE SECURITY ENHANCEMENTS – PART 1

*Inderpal S. Johal, Data Softech Inc.*

## INTRODUCTION

Oracle 11g has introduced lots of security features. In this paper, I will discuss some of the security features associated with User, auditing etc.

## USER PASSWORD – CASE SENSITIVE

In Oracle Database 11g, the password is handled differently than in previous versions. Passwords are case sensitive in 11g.

1. When the database is created using dbca, the passwords will be case sensitive by default.
2. When an Oracle Database 10g is upgraded, passwords are case insensitive until the ALTER USER... command is used to change the password.

```

ORACLE 10g
DATABASE
SQL> create user indy identified by INDY;
User created.

SQL> conn indy/indy      →Not Case-Sensitive
Connected.

ORACLE 11g
DATABASE
SQL> create user indy identified by INDY;
User created.

SQL> conn indy/indy
ERROR:
ORA-01017: invalid username/password; logon denied
Warning: You are no longer connected to ORACLE.

SQL> connect indy/INDY
Connected.

You can make it Case in-sensitive as in previous version
SQL> show parameter sensitive
NAME                                TYPE          VALUE
-----
sec_case_sensitive_logon            boolean       TRUE

SQL> alter system set sec_case_sensitive_logon=false;
System altered.

SQL> alter user indy identified by INDY;
User altered.

SQL> conn indy/indy
Connected.

```

## USER PASSWORD – DBA\_USERS

You cannot see the password in DBA\_USERS and so do you thing that we are not allowed to use the old sql ALTER USER .. IDENTIFIED BY VALUES to change the user password.

```
SQL> select username,password from dba_users where username=' INDY' ;
```

```
USERNAME                                PASSWORD
-----                                -
```

```
INDY
```

```
SQL> select name,password from user$ where name=' INDY' ;
```

```
NAME                                PASSWORD
-----                                -
```

```
INDY                                8C477A91C1AF4CE3
```

So don' t worry we can continue to use the same old methodology to revert back the User password. The Null password in DBA\_USERS is due to change in the View definition as shown below



```
$ORACLE_HOME/rdbms/admin/catalog.sql
```

```
create or replace view DBA_USERS
```

```
  (USERNAME, USER_ID, PASSWORD, ACCOUNT_STATUS, LOCK_DATE, EXPIRY_DATE,
   DEFAULT_TABLESPACE, TEMPORARY_TABLESPACE, CREATED, PROFILE,
   INITIAL_RSRC_CONSUMER_GROUP, EXTERNAL_NAME)
```

```
as
```

```
select u.name, u.user#, u.password,m.status, .....
```



```
$ORACLE_HOME/rdbms/admin/cdenv.sql
```

```
create or replace view DBA_USERS
```

```
  (USERNAME, USER_ID, PASSWORD, ACCOUNT_STATUS, LOCK_DATE, EXPIRY_DATE,
   DEFAULT_TABLESPACE, TEMPORARY_TABLESPACE, CREATED, PROFILE,
   INITIAL_RSRC_CONSUMER_GROUP, EXTERNAL_NAME, PASSWORD_VERSIONS,
   EDITIONS_ENABLED)
```

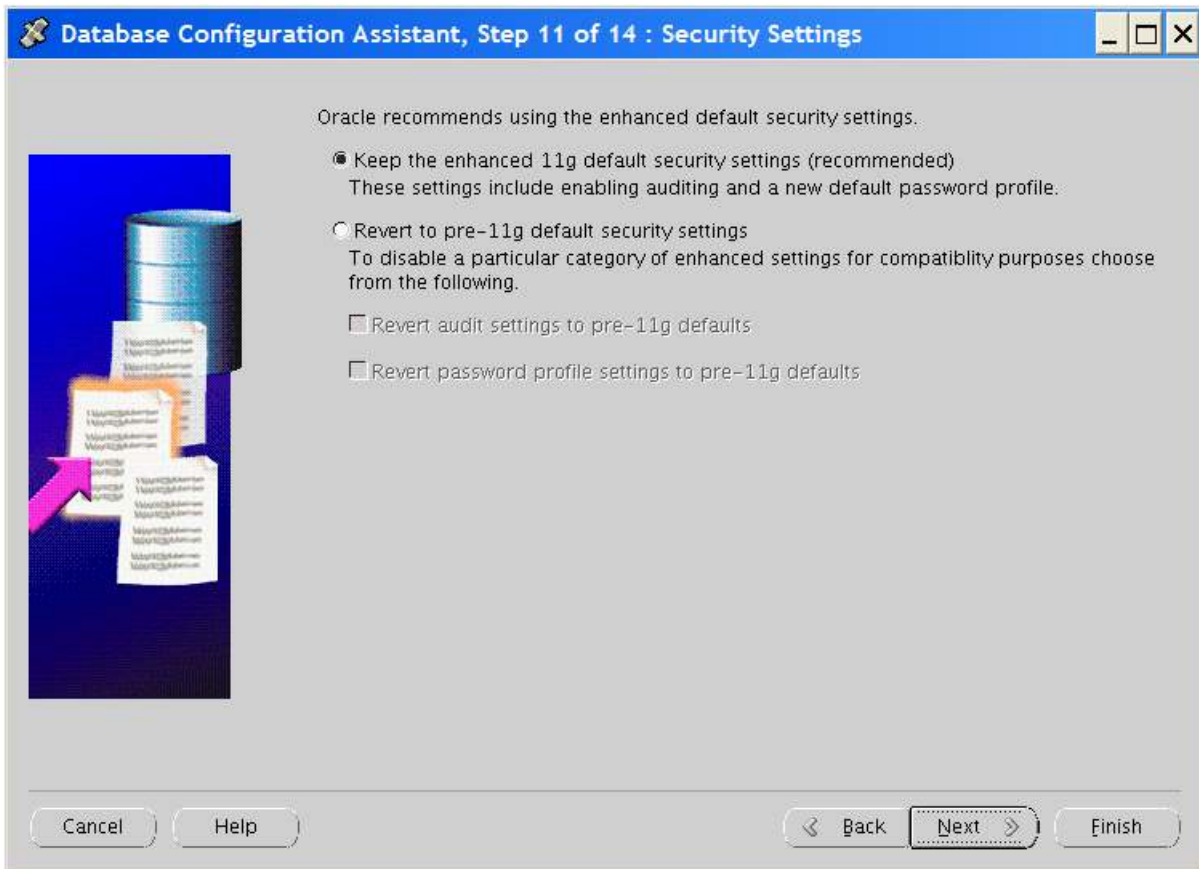
```
as
```

```
select u.name, u.user#,
       decode(u.password, 'GLOBAL', u.password,
               'EXTERNAL', u.password,
               NULL),
       m.status,
```

## **ORACLE 11G NEW SECURITY SETTING SCREEN**

When you are creating new Database using dbca, you will see the new Security Setting screen as shown below. You have two options and it will be discussed in details in the next few pages in this paper

1. Keep the 11g Default Security settings like Auditing Enabled at DB level and Password profile
2. Revert to pre-11g Default Security setting. Here you will still have the choice to enable either of the
  - a. Revert only Auditing to pre-11g defaults
  - b. Revert only Password profile to pre-11g defaults



## USER PASSWORD – DEFAULT PASSWORD PROFILE

When creating a custom database using the Database Configuration Assistant (DBCA), you can specify the Oracle Database 11g default security configuration. Account is locked after 10 failed login attempts. Moreover if a user tries to connect to 11g database with an incorrect password, the instance delays each login after the third try. This protection even applies for attempts made from different IP addresses.. After 3 unsuccessful, Oracle gradually increases the time before the user can try another password, up to a maximum of about ten seconds. See an example below where I am trying to connect to the database with wrong password

```
$ cat a.sh
sqlplus -s /nolog <<EOF
conn indy/indyl
exit;
EOF
$ while true ; do ./a.sh; date; done
ERROR:
ORA-01017: invalid username/password; logon denied

Fri Aug 17 13:09:48 EDT 2007 → 1st Attempt after 0 second
ERROR:
ORA-01017: invalid username/password; logon denied

Fri Aug 17 13:09:48 EDT 2007 → 2nd Attempt after 0 second
ERROR:
ORA-01017: invalid username/password; logon denied

Fri Aug 17 13:09:48 EDT 2007 → 3rd Attempt after 0 second
ERROR:
ORA-01017: invalid username/password; logon denied

Fri Aug 17 13:09:49 EDT 2007 → 4th Attempt after 1 second
ERROR:
ORA-01017: invalid username/password; logon denied

Fri Aug 17 13:09:51 EDT 2007 → 5th Attempt after 2 second
ERROR:
ORA-01017: invalid username/password; logon denied

Fri Aug 17 13:09:55 EDT 2007 → 6th Attempt after 4 second
ERROR:
ORA-01017: invalid username/password; logon denied

Fri Aug 17 13:09:59 EDT 2007 → 7th Attempt after 4 second
ERROR:
ORA-01017: invalid username/password; logon denied

Fri Aug 17 13:10:04 EDT 2007 → 8th Attempt after 5 second
ERROR:
```

```
ORA-01017: invalid username/password; logon denied

Fri Aug 17 13:10:10 EDT 2007 → 9th Attempt after 6 second
ERROR:
ORA-01017: invalid username/password; logon denied

Fri Aug 17 13:10:17 EDT 2007 → 10th Attempt → Account Locked → 10 second after 1st attempt
ERROR:
ORA-28000: the account is locked
```

The default password profile is enabled with these settings at database creation:

```
PASSWORD_LIFE_TIME 180
PASSWORD_GRACE_TIME 7
PASSWORD_REUSE_TIME UNLIMITED
PASSWORD_REUSE_MAX UNLIMITED
FAILED_LOGIN_ATTEMPTS 10
PASSWORD_LOCK_TIME 1
PASSWORD_VERIFY_FUNCTION NULL
```

## USER PASSWORD – BUILT –IN PASSWORD COMPLEXITY CHECKER

Prior to Oracle 11g, we can use `verify_function` function in `utlpwdmg.sql` to enable Password complexity. In Oracle 11g, `utlpwdmg` has another new function named `verify_function_11g`, which will enforce the following password restriction.

```
SQL> show user
USER is "SYS"
SQL> @?/rdbms/admin/utlpwdmg
Function created.
Profile altered.
Function created.

SQL> alter profile default
  2 limit
  3 password_verify_function verify_function_11g;

Profile altered.
```

You can select any of these function and can customize as per your requirements.

### 1. Password must contains at least 8 characters

```
SQL> create user indy identified by indy;
create user indy identified by indy
*
ERROR at line 1:
ORA-28003: password verification for the specified password failed
ORA-20001: Password length less than 8
```

### 2. Password must contains at least one number and one alphabetic character, and differs from the previous password by at least 3 characters.

```
SQL> create user indy identified by inderpal;
create user indy identified by inderpal
*
ERROR at line 1:
ORA-28003: password verification for the specified password failed
ORA-20008: Password must contain at least one digit, one character8
```

### 3. Password is not same like

- username or
- username appended with a number 1 to 100.,
- username reversed,
- server name or
- server name appended with 1-100, or
- Common easily guessed password like 'welcome1', 'database1'

```
SQL> create user inderpal identified by inderpal1;
create user inderpal identified by inderpal1
*
ERROR at line 1:
ORA-28003: password verification for the specified password failed
ORA-20005: Password same as or similar to user name
```

## **MANAGING ORACLE 11G DEFAULT AUDITS**

In Oracle 11g, auditing is enabled by default for certain privileges w.r.t security. The audit trail is by default recorded in the database AUD\$ table which can impact the performance of the large database. It is recommended to move the auditing from database to OS audit trail files.

### *LIST OF PRIVILEGES WHICH ARE **AUDITED** BY DEFAULT IN 11G*

```
CREATE EXTERNAL JOB
CREATE ANY JOB
GRANT ANY OBJECT PRIVILEGE
EXEMPT ACCESS POLICY
CREATE ANY LIBRARY
GRANT ANY PRIVILEGE
DROP PROFILE
ALTER PROFILE
DROP ANY PROCEDURE
ALTER ANY PROCEDURE
CREATE ANY PROCEDURE
ALTER DATABASE
GRANT ANY ROLE
CREATE PUBLIC DATABASE LINK
DROP ANY TABLE
ALTER ANY TABLE
CREATE ANY TABLE
DROP USER
ALTER USER
CREATE USER
CREATE SESSION
AUDIT SYSTEM
ALTER SYSTEM
```

## USING TABLESPACE ENCRYPTION

Oracle10g Rel2 has introduced a new feature called Transparent Data Encryption (TDE). TDE is beneficial for simple and easy encryption of sensitive data in table columns. Simple and easy because users or applications need not manage the encryption and decryption of data any more, it is handled by the database - so there is no need to manage views, tables, or triggers to decrypt data.

The encryption keys are stored in a location (file named ewallet.p12) external to the database. This location can be either OS-specific or location specified in sqlnet.ora file using parameter WALLET\_LOCATION.

TDE can be used to protect confidential data like credit card information, social security number, etc. The data encrypted by TDE cannot be accessed unless authorized decryption occurs, which is automatic for users authorized to access the tables. For example, even if the disks are stolen, the data is safe as the encrypted table columns cannot be viewed unless the master key (stored in ewallet.p12) is provided. The master key is password provided while creating the wallet.

### *ENABLING TDE*

```
SQL> show user
USER is "SYS"

SQL> alter system set encryption key identified by "ips123xyz";
alter system set encryption key identified by "ips123xyz"
*
ERROR at line 1:
ORA-28368: cannot auto-create wallet

SQL> !mkdir $ORACLE_BASE/admin/$ORACLE_SID/wallet

SQL> alter system set encryption key identified by "ips123xyz";
System altered.

SQL> !ls -ltr $ORACLE_BASE/admin/$ORACLE_SID/wallet
total 4
-rw-r--r-- 1 oracle oinstall 1693 Aug 19 15:15 ewallet.p12
```

Oracle 11g Tablespace encryption is based on block level encryption and so data encryption is not occurring in the memory. Data encrypts occurs on write and decrypts on read. The only encryption penalty is associated with I/O. This encryption has no impact on Queries access path and supports all data types of the database.

```
SQL> create tablespace tde_tblspc
 2 datafile '/home/oracle/app/oradata/orcl/tde_tblspc' size 50M
 3 encryption using 'AES128'
 4 default storage (encrypt);
Tablespace created.
```

In above Syntax,

ENCRYPTION clause sets the Encryption algorithm. Valid Encryption algorithms are

- 3DES168
- AES128 [Default]
- AES192
- AES256.

ENCRYPT storage clause cause the encryption to be used.

```

You can get the information about the Encryption defined on tablespace level as shown
below
SQL> desc v$encrypted_tablespaces
Name                                     Null?    Type
-----
TS#                                       NUMBER
ENCRYPTIONALG                          VARCHAR2(7)
ENCRYPTEDTS                               VARCHAR2(3)

SQL> select * from v$encrypted_tablespaces;
TS# ENCRYPT ENC
-----
7 AES128 YES

```

The encrypted data is protected during operations like JOIN and SORT. This means that the data is safe when it is moved to temporary tablespaces.

Data in undo and redo logs is also protected.

Encrypted tablespaces are transportable if the platforms have same endianness and the same wallet.

**RESTRICTIONS**

1. You cannot encrypt Temporary and Undo Tablespaces.
2. Bfiles and external tables are not encrypted.
3. The key for an encrypted tablespaces cannot be changed at this time. The only available solution is create a new tablespace with the desired properties and move all objects to the new tablespace.