Step-By-Step Guide to Installing and Implementing Oracle 11g Enterprise User Security

Mike Dean
Sr Staff Consultant
Database Specialists, Inc
http://www.dbspecialists.com
mdean@dbspecialists.com

Managing database user accounts in a large enterprise can be very challenging. Creating individual accounts for every person in every database can be very time-consuming for DBAs and managing different passwords in every database is time-consuming for both DBAs and users. When people leave the enterprise, it is critical to remove their access immediately but this can be difficult if you have many databases. However, the cost of not properly managing these accounts is that your databases are more vulnerable to security breaches. Oracle offers a solution to this in the form of Enterprise User Security (EUS) which is part of the Oracle Identity Management product. With EUS, user accounts are created and managed in an LDAP repository called Oracle Internet Directory (OID). OID is responsible for authenticating the users as well as managing the access rights of each user.

In this paper, I will give step-by-step instructions on how to install the software components required to run Oracle Identity Management and also how to implement Enterprise User Security to handle a typical real-world scenario in which different people need different levels of access to a number of databases.

1) install 11.2.0.3 database software

2) create database called OIDREP

3) Install Weblogic Server10.3.2 binaries
/usr/java/jdk1.7.0_03/bin/java –d64 –jar wls1032_generic.jar
(downloaded the 64-bit version from technet.oracle.com)
Middleware Home : /opt/oracle/product/Middleware
Typical Install
WebLogic Server Home: /opt/oracle/product/Middleware/wlserver_10.3
Don't run QuickStart

4) upgrade to WLS 10.3.5 with patch 12395574
unzip p12395574_1035_Generic.zip
(downloaded from Oracle Support)
/usr/java/jdk1.7.0_03/bin/java –d64 –jar wls1035_upgrade_generic.jar
Install into existing Middleware Home
Uncheck Oracle Coherence
Don't run QuickStart

5) Install Oracle Identity Management 11.1.1.2
unzip ofm_idm_idm_linux_11.1.1.2.0_64_disk1_1of1.zip
(downloaded from technet.oracle.com)

```
cd Disk1
./runInstaller
```

Install Software – Do Not Configure

Oracle Home Directory: Oracle_IDM1
Run Oracle_IDM1/oracleRoot.sh as root

6) upgrade Identity Management to to 11.1.1.5 using patch 12395123
unzip p12395123_111150_Linux-x86-64.zip
(downloaded from Oracle Support)
```
cd Disk1
./runInstaller
```
Run Oracle_IDM1/oracleRoot.sh as root
Answer Yes to question about Privileged ports

7) Create a New Domain using the IDM Configuration Wizard
```
cd /opt/oracle/Middleware/Oracle_IDM1/bin
./config.sh
```
Create New Domain
Username/Password: weblogic/oracle123
Domain Name: IDMDomain
Uncheck Oracle Identity Federation
Create Schema
Connect String: mdlinux:1521:OIDREP

8) Check that OID is up and running
```
ldapbind -h mdlinux -p 3060
ldapbind -h mdlinux -p 3131 -U 1
```
(should each say “bind successful”)

9) Check that WebLogic Server is running
Open http://mdlinux:7001/console with a browser and login as weblogic

10) Configure OID to accept AnonymousBinds
```
ldapmodify -D cn=orcladmin -q -p 3060 -h mdlinux -f /tmp/ldif.txt
```

/tmp/ldif.txt:
```
dn: cn=oid1,cn=osdldapd,cn=subconfigsubentry
changetype: modify
replace: orclAnonymousBindsFlag
orclAnonymousBindsFlag: 1
```

11) Use NetCA to configure the Oracle Home for directory usage.
```
./netca
```
Directory Usage Configuration
Directory Type: Oracle Internet Directory
Hostname: mdlinux
Port: 3060
SSL Port: 3131
cn=OracleContext
Finish

This will create an $ORACLE_HOME/network/admin/ldap.ora:

```
DIRECTORY_SERVERS= (mdlinux:3060:3131)
DEFAULT_ADMIN_CONTEXT = "dc=localdomain"
DIRECTORY_SERVER_TYPE = OID
```

(Note: Make sure that the DEFAULT_ADMIN_CONTEXT is populated correctly. Otherwise the following steps won't work correctly)

12) Use DBCA to register the database(s) with the OID

I created a database called OIDREP to use as the OID Repository. I also created OIDTEST which I will use going forward. The OID Repository should be kept isolated in its own database, separate from the ones that are using OID.

```
./dbca
```

Configure Database Options
Select OIDREP
Yes, Register the Database
User DN: cn=orcladmin
Specify a Wallet Password

Do the same thing for OIDTEST. This does: "alter system set ldap_directory_access='PASSWORD' scope=both;". It also creates an entry for each database in OID under dc=localdomain called cn=OracleContext

13) Confirm databases are registered using Enterprise User Security Manager (eusm)

```
eusm listDomainInfo domain_name="OracleDefaultDomain"
realm_dn="dc=localdomain" ldap_host="mdlinux" ldap_port="3060"
ldap_user_dn="cn=orcladmin" ldap_user_password="oracle123"
DOMAIN INFOMATION FOR DOMAIN:
OracleDefaultDomain
-----------------------------------------------
Current user DB links status: DISABLED
Allowed user authentication methods: ALL
LIST OF DATABASES
-----------------
oidrep
oidtest
```

So at this point, Oracle Identity Management should be up and running and ready for use. We will now implement EUS using a shared schema and enterprise roles. Users do not necessarily require individual accounts or schemas set up in each database. Alternatively, they can connect to a shared schema and be granted access to the objects associated with target applications. For example, suppose that users Tom, Dick, and Harriet require access to the Payroll application on the Finance database. They do not need to create unique objects in the database, and therefore do not need their own schemas, but they do need access to the objects in the Payroll schema.

Oracle Database supports mapping multiple users stored in an enterprise directory to a shared schema on an individual database. This separation of users from schemas reduces administration costs by reducing the number of user accounts on databases. It means that

you do not need to create an account for each user (user schema) in addition to creating the user in the directory. Instead, you can create a user in the enterprise directory, and map that user to a shared schema. Other enterprise users can also be mapped to that schema.

For example, if Tom, Dick and Harriet all access both the Sales and the Finance databases, you do not need to create an account for each user on each database. Instead, you can create a single shared schema on each database, such as `GUEST`, that all three users can access. Then individual access to objects in the Sales or Finance database can be granted to these three users by using enterprise roles. A typical environment can have up to 5,000 enterprise users mapped to one shared schema and each user can be assigned a set of enterprise roles.

Oracle recommends that you create a separate shared schema that contains no objects to use as an entry point. Then, grant access to application objects in other schemas through enterprise roles. Otherwise, application objects can be inadvertently or maliciously deleted or altered.

In summary, shared schemas provide the following benefits:

- Shared schemas eliminate the need to have a dedicated database schema on each database for each enterprise user.
- Each enterprise user can be mapped to a shared schema on each database the user needs to access. The user connects to the shared schema when the user connects to a database.
- Shared schemas lower the cost of managing users in an enterprise.

So now I will show you how to implement Enterprise User Security in a real-world scenario using shared schemas. Imagine a company in which there are 3 different groups in the IT department that support a particular application: Developers, Analysts and Application DBAs. The Developers need to be able to read from the application schema as well as create their own tables. The Analysts simply need to read from the application schema and the Application DBA needs full read-write privileges on the application schema. For the sake of simplicity, this application schema consists of a single table called “application_table”. Joe is the Developer, Tom is the Analyst and Sue is the Application DBA.

1) Use Oracle Directory Services Manager (ODSM) to create three new users: Joe, Tom and Sue

Go to http://mdlinux.localdomain:7006/odsm/faces/odsm.jspx
Click on Connect to a Directory
Directory Type: OID
Name: IDMDomain

Server: mdlinux
Port: 3060
Do not check SSL enabled
Log in with Username: cn=orcladmin
Click on Data Browser
Expand dc=localdomain
Expand cn=Users
Right Click on cn=orcladmin
Create Like
Next
Fill in the Mandatory Properties:
Cn: Joe
Sn: Joe
Relative Distinguished Name: cn
Distinguished Name: cn=Joe,cn=Users,dc=localdomain
Next
Under Optional Properties, change "orcladmin" to Joe and fill in UserPassword

Do this same procedure for Tom and Sue

2) Create a shared schema
Now we will create a global shared schema in the database.

```
SQL> create user shared_schema identified globally;
SQL> grant connect to shared_schema;
```

3) Map the Enterprise Users to the shared schema using DB Console
Log into DB Console for the OIDREP database at http://mdlinux:1158/em

Click on Server
Enterprise User Security
Log in as User: cn=orcladmin,cn=users,dc=localdomain
Click on Login
Manage Enterprise Domains
Configure OracleDefaultDomain
User-Schema mappings
Click on Create
Subtree
cn=users,dc=localdomain
Schema: shared_schema
Click on Continue
OK

By granting access at the SubTree level, all users at the cn=Users level are mapped to the shared schema. If you don't select SubTree, then you will need to map each user individually.

4) Test the shared schemas

```
SQL> connect joe/joe12345
Connected.
SQL> show user
USER is "SHARED_SCHEMA"
SQL> connect tom/tom12345
Connected.
SQL> show user
USER is "SHARED_SCHEMA"
SQL> connect sue/sue12345
Connected.
SQL> show user
USER is "SHARED_SCHEMA"
```

So at this point, Joe, Tom and Sue can log into the database but they don't have any privileges except CREATE SESSION. Notice that the database user is "SHARED_SCHEMA" even though they each log in with their own account.

Enterprise Users can be identified in the Audit Trail by the COMMENT_TEXT column specifying how they were authenticated.

```
SQL> select COMMENT_TEXT from dba_audit_trail;

Authenticated by: DIRECTORY PASSWORD;EXTERNAL NAME:
cn=joe,cn=Users,dc=localdomain
```

So now we will create Global Database Roles and Enterprise Roles to manage their privileges.

5) Create the Global Database Roles

```
Create role DEVELOPER identified globally;
Create role ANALYST identified globally;
Create role APP_DBA identified globally;
```

Create the regular database roles and grant necessary privileges

```
Create role APP_READ_ONLY;
Grant select on app_owner.application_table to APP_READ_ONLY;
Create role APP_READ_WRITE;
Grant select, insert, update, delete on app_owner.application_table to
APP_READ_WRITE;
```

Grant the Roles and any extra privileges (create table) to the Global Database Roles

```
Grant create table, APP_READ_ONLY to DEVELOPER;
Grant APP_READ_ONLY to ANALYST;
Grant APP_READ_WRITE to APP_DBA;
```

Notice that the SHARED_SCHEMA user doesn't have any grants, other than CREATE SESSION. All grants are given to database roles, which are in turn granted to the global roles.

6) Use DB Console to create 3 corresponding Enterprise Roles: Developer, Analyst and Application DBA and assign the proper person to the proper role.

Log into DBConsole, click on Server, Enterprise User Security
Manage Enterprise Domains
Configure OracleDefaultDomain
Enterprise Roles
Create
Put in "Developer" for the name of the Enterprise Role and click Add
Select oidtest from dropdown
Username: System
Go
Select DEVELOPER
Continue
OK
Configure OracleDefaultDomain
Enterprise Roles
Edit Developer
Grantees
Add
Go
Select Joe
OK
Continue

Do this again for Tom and Sue, adding them to the Analyst and Application DBA Enterprise Roles. Now test everyone's privileges. Remember that Tom should have read-only, Joe has read-only plus create table and Sue has read-write.

```
SQL> conn tom/tom12345
Connected.

SQL> select * from app_owner.application_table;

      1

SQL> delete from app_owner.application_table;
delete from app_owner.application_table
                      *
ERROR at line 1:
ORA-01031: insufficient privileges


SQL> create table tab1 (col1 number);
```

```
create table tab1 (col1 number)
*
ERROR at line 1:
ORA-01031: insufficient privileges


SQL> conn sue/sue12345
Connected.
SQL> select * from app_owner.application_table;
         1

SQL> delete from app_owner.application_table;
1 row deleted.

SQL> create table tab1 (col1 number);
create table tab1 (col1 number)
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> conn joe/joe12345
Connected.
SQL>  select * from app_owner.application_table;
no rows selected

SQL> delete from app_owner.application_table;
 delete from app_owner.application_table
                       *
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> create table tab1 (col1 number);
Table created.
```

Everything now works as expected. But you may be wondering what the advantage to doing all of this is. So far, it seems like a lot of extra work with no real benefit.

Suppose the IT department with 3 people and one database expands to 3000 people and 100 databases. Without Enterprise Security, you would need to (theoretically) create 3000 individual accounts in each of the 100 databases. With Enterprise Security, for each new database, all you need to do is register it with OID (See Step 12 above), create the shared schema and create the roles. Once you do that, all Enterprise Users will have access and the appropriate privileges. There is no need to create individual user accounts as they are all kept in the Internet Directory. With just three users, it is not much trouble, but with 3000 users and 100 databases, this would obviously take a long time.

Another advantage comes when the 3000 users start changing their passwords. Without Enterprise Security, they will need to change their passwords separately in each of the 100 databases. With Enterprise Security, they can use the Self-Service Console or Oracle Directory Services Manager and change it themselves in the directory. Oracle Internet Directory Self-Service Console is a tool based on Delegated Administration Services. This is a self service application that allows administrated access to the applications data

managed in the directory. This tool comes ready to use with Oracle Internet Directory 10g but for some reason does not come as part of OID 11g so you need to install the 10g version in OID 11g. The Oracle Identity Management Guide to Delegated Administration discusses Delegated Administration Services and the Oracle Internet Directory Self-Service Console tool.

When someone leaves the company, their database access can be removed simply by updating their account in the OID rather than in each database. If someone changes roles from Developer to Analyst, you can change their privileges in OID rather than in each database.

So, as you can see, there is certainly extra effort involved in setting up and maintaining Enterprise User Security compared to traditional database users. As the number of databases and users that you have to manage goes up, the benefits of EUS may start to outweigh the costs. Whether or not it makes sense for your enterprise would need to be carefully considered.

Appendix

These are some Oracle Support documents that I found useful while doing this research:
Troubleshooting Enterprise User Security [ID 191137.1]

How to Debug Problems with Enterprise User Security [ID 398524.1]

Step by Step Guide To Troubleshooting 10g Enterprise User Security (EUS) - Password Authentication [ID 453853.1]

EUSM, Command Line Tool For EUS Administration and Some EUS Good to Knows [ID 1085065.1]

Troubleshooting OID 11g and Later [1347487.2]

INST-07527 Error When Patching Oracle Fusion Middleware 11g To Higher Version [ID 1261856.1]

Steps to Maintain Oracle Fusion Middleware 11g Release 1 (11.1.1) [ID 1073776.1]

How Can I Download Upgrade Installers for Oracle WebLogic Server (WLS)? [ID 1074946.1]

Information Center: Oracle Internet Directory 11g and later [ID 1346624.2]

How to Start a WebLogic 10.3.x Domain AdminServer Without Interactively Supplying the Username / Password? [ID 980292.1]

11g Identity Management documentation:
http://docs.oracle.com/cd/E27559_01/index.htm

This is a script that I use to manually start the components of Oracle Identity Management, including the OID Repository and listener.

```
export ORACLE_SID=OIDREP
export ORAENV_ASK=NO
export
IDM_HOME=/opt/oracle/product/Middleware/user_projects/domains/IDMDomain
. oraenv
lsnrctl start
sleep 30
sqlplus / as sysdba <<EOF
startup
EOF

# start the WebLogic Admin Console (port 7001)
echo "starting WebLogic Admin Server"
nohup $IDM_HOME/bin/startWebLogic.sh >/dev/null 2>/dev/null &
sleep 130
# start the Oracle Directory Services application (port 7006)
echo "Starting Oracle Directory Services Manager"
nohup $IDM_HOME/bin/startManagedWebLogic.sh wls_ods1 >/dev/null
2>/dev/null &

echo "Starting Oracle Internet Directory"
cd /opt/oracle/product/Middleware/asinst_1/bin
./opmnctl startall
sleep 130
./opmnctl status -l
```