Product Profiles – a forgotten feature in Oracle

Product Profiles have been available in Oracle for many years. The supporting objects are installed in most databases, but are very seldom used.
This article outlines the advantages of using Product Profiles as a security tool within a database. The set up and maintenance requirements are described.
Product Profiles provides a flexible way to restrict what users can do in a database. As it is part of the Oracle toolset and has no extra cost associated with it, there should be very little reason not to implement it.

After installing Oracle and creating a new database, DBAs are greeted with the following messages the first time they run sqlplus:

```
Error accessing PRODUCT_USER_PROFILE
Warning:  Product user profile information not loaded!
You may need to run PUPBLD.SQL as SYSTEM
```

This warning often goes un-noticed, until someone actually reads it and asks about it. Typically the DBA will then find the pupbld.sql script and run it. The message disappears and everyone is happy.

But why do we need to run this script, and what does it give us?

Product Profiles have been available in Oracle for many versions, but are not often used, and are not mentioned in many reference books. This is unfortunate, as they provide a convenient, no cost way to restrict what users can do by disabling specified (SQL, PL/SQL or sqlplus) commands or roles for specified (groups of) users within SQL*Plus.

The PUPBLD.SQL script creates a number of objects. Of interest is the table sqlplus_product_profile which has 8 columns, of which only 4 are used by SQL*Plus. These columns are:
PRODUCT, set to the constant value ‘SQL*PLUS’;
CHAR_VALUE, set to ‘DISABLED’, or the name of a role;
USERID, set to the name of the user (or a pattern match for a group of users); and
ATTRIBUTE, contains the command that is to be disabled or the fixed value ‘ROLES’.

```
create table sqlplus_product_profile
(
  product        varchar2 (30) not null,
  userid         varchar2 (30),
  attribute      varchar2 (240),
  scope          varchar2 (240),
  numeric_value  decimal (15,2),
  char_value     varchar2 (240),
  date_value     date,
  long_value     long
);
```

As an example, consider the following (not untypical) situation:

> The installation procedure for a purchased application requires the creation of a user (APPOWNER identified by APPOWNER) with DBA role. The application connects to the database using this (hard-coded) username and password. Any modification to this will prevent the application from working (and invalidate your support!).

It is possible for any user to log into the database using the APPOWNER account, and in doing so, gain DBA access to the database. This is probably not what most DBAs would want in a production environment, as the user has the ability to destroy the database.

By using Product Profiles, it is possible to restrict what a user can do within sqlplus. For example, it is possible to disable the SQL “DROP” command for the APPOWNER user by entering the following:

```
INSERT INTO SQLPLUS_PRODUCT_PROFILE (PRODUCT, USERID, ATTRIBUTE, CHAR_VALUE) VALUES
('SQL*PLUS','APPOWNER','DROP','DISABLED');
```

Additionally, the DBA role can be disabled by entering:

```
INSERT INTO SQLPLUS_PRODUCT_PROFILE (PRODUCT, USERID, ATTRIBUTE, CHAR_VALUE) VALUES
('SQL*PLUS', 'APPOWNER', 'ROLES', 'DBA');
```

These restrictions are enforced by sqlplus, not the database, and will therefore not impact the functioning of the application. They are also enforced after the user has logged in, meaning that the user will still be able to log into the database (a privilege granted through DBA role) but once logged in, all privileges granted through the DBA role (e.g. “Drop any table”) will be disabled.

The table below lists commands that can be disabled through the use of Product Profiles. As can be seen, this is a fairly comprehensive list of commands. This, together with the ability to disable any role that the user may have access to, forms a very powerful extension to any existing security that may already be in place.

ALTER	ANALYZE	AUDIT
BEGIN	CONNECT	CREATE
DECLARE	DELETE	DROP
EDIT	EXECUTE	EXIT
GET	GRANT	HOST
INSERT	LOCK	NOAUDIT
QUIT	RENAME	REVOKE
RUN	SAVE	SELECT
SET	SET ROLE	SET TRANSACTION
SPOOL	START	TRUNCATE
UPDATE		

Table 1 Commands that can be disabled with Product Profiles.

About the author:
Andrew Deighton is a database consultant and technical director at Labyrinth IT Services Ltd. He has been working with Oracle for more than 10 years and is an Oracle Certified 8i DBA. He has published several papers on various topics related to Oracle. He also works as an instructor for Learning Tree International. He can be contacted at andrew@labyrinth-it.co.uk.