

Oracle Operating System Authentication (How to use operating system authentication)

By Mark E. Donaldson

For a long time, the Oracle database on most platforms supported operating system (OS) authentication. In normal authentication, a user specifies a username and password to connect to a database. In these scenarios, most database users and applications end up sharing the same account with all the same privileges and restrictions. With OS authentication, the database allows a user to connect to a database without a password, accepting the OS name as the database username.

For example, if I log into a UNIX host as scott, I can connect to the database, using SQL*Plus by simply typing:

```
sqlplus /
```

OS-authenticated accounts are normal database accounts that start with the prefix defined by the OS_AUTHENT_PREFIX in the database init.ora parameters by default OP\$\$\$. In the default case, the OS user scott would be connected to the database as the database user OP\$\$\$SCOTT. To create the user, you could use a command like:

```
create user ops$$scott identified by rabbit default tablespace users  
temporary tablespace temp;
```

or

```
create user ops$$scott identified externally default tablespace users  
temporary tablespace temp;
```

In the first format, a specific password is assigned to the OS account's log in, so users can still connect to the database using ops\$\$scott/rabbit even if they aren't connected through to the OS account. The second format means that there is no password, and users can only connect if they're in that OS account on the same machine.

There are several reasons for DBAs to prefer OS authentication:

- OS-authenticated accounts are easy to monitor, track, and restrict on a per-user basis.
- You can change or hide a schema password from end users without affecting accounts connecting by OS authentication.
- If a user's OS account is revoked, they wouldn't be able to get into the database through another account on the same machine unless they knew the OS authenticated account password--that's if the database account was even still valid.
- Database maintenance scripts and programs can be written without hard-coding or prompting for username/password accounts.

OS authentication was used quite a bit during the days of character-mode terminals and central hosts, but it has fallen off due to the trend toward network distributed applications and Web servers. There is an alternative OS authentication method, where the database will attempt to authenticate a user by their remote username by using the REMOTE_OS_AUTHENT init.ora parameter, which can be set to either TRUE or FALSE. It's FALSE by default because it's very vulnerable to attack.

Oracle Operating System Authentication

(How to use operating system authentication)

By Mark E. Donaldson

Any user who can set up a machine, even on a PC, and knows how to set their local username can connect to a database without a password using `/@dbname`. To prevent this vulnerability, you can use a `protocol.ora` file, setting `tcp.invited_nodes` to a list of IP addresses that are allowed to connect remotely and `tcp.validnode_checking=yes` to verify remote OS-authenticated accounts against the list of IP addresses. In environments where most connections are going through a Web server, there might only need to be a short list of trusted hosts.