CHAPTER 8

Event 10079 and Oracle Net Packet Contents

The undocumented event 10079 may be used to dump Oracle Net traffic to a trace file. It is useful for quickly determining which SQL statements, PL/SQL calls, or SQL*Plus commands send sensitive data such as passwords unencrypted.

Event 10079 is similar to Oracle Net tracing in that it dumps the complete contents of network packet contents between database client and server. It is more convenient than changing sqlnet.ora to enable dumping of Oracle Net packet contents. Unlike trace_level_client in sqlnet.ora, it may also be used to enable packet dumps for database sessions that are already established. Table 8-1 lists the supported event levels.

Table 8-1. Supported Levels of Event 10079

<table>
<thead>
<tr>
<th>Level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Trace network operations to/from client</td>
</tr>
<tr>
<td>2</td>
<td>In addition to level 1, dump data</td>
</tr>
<tr>
<td>4</td>
<td>Trace network operations to/from database link</td>
</tr>
<tr>
<td>8</td>
<td>In addition to level 4, dump data</td>
</tr>
</tbody>
</table>

Case Study

The subsequent sections assume that the Advanced Security Option for the encryption of Oracle Net traffic is not used. The SQL*Plus User’s Guide and Reference does not state whether passwords are sent encrypted when modified with the SQL*Plus command PASSwORD. Event 10079 may be used to find out.
SQL> CONNECT / AS SYSDBA
Connected.
SQL> ALTER SESSION SET EVENTS '10079 trace name context forever, level 2';
Session altered.
SQL> PASSWORD ndebes
Changing password for ndebes
New password:
Retype new password:
Password changed
SQL> ORADEBUG SETMYPID
Statement processed.
SQL> ORADEBUG TRACEFILE_NAME
/opt/oracle/obase/admin/TEN/udump/ten1 ora_20364.trc

The resulting trace file contains the following packet dump:

```
C850B0  FF6686BF  646E06BF  73656265  00000030  [.h....ndebes0...]
C850B0  54554110  454E5F48  53415044  524F5700  [.AUTH_NEWPASSWORD]
C850B0  0000C044  31384000  39314642  38373930  [D....@81F190978]
C850C0  41323232  39363642  45453539  42303242  [222AB66995EB20B]
C850C0  46323546  30324343  30313239  39453434  [F52FCC20921044E9]
C850C0  34423130  32353232  45423332  44393431  [01B4225223E149D]
C850C0  30304542  00003245  00270000  410D0000  [BEE0E2....'....A]
C850C0  5F485455  53534150  44524F57  00000000  [UTH_PASSWORD....]
```

Obviously, the password was sent encrypted. Thus, the SQL*Plus PASSWORD command is a safe way to change passwords, whereas ALTER USER user_name IDENTIFIED BY new_password is not, since it sends the password unencrypted along with the SQL statement text. By the way, the preceding encryption is different from the password hash in DBA_USERS.PASSWORD, such that eavesdropping a communications link cannot be used to glean password hashes stored in the dictionary base table USER$. Oracle Call Interface provides the function OCIPasswordChange() and it is safe to assume that SQL*Plus uses this function to implement the PASSWORD command. Unfortunately the manuals do not state whether or not OCIPasswordChange() encrypts passwords.

Some applications use roles, which are protected by a password, to enable certain privileges only when a user connects with the application. This is intended to restrict the privileges of users who connect with SQL*Plus or other applications. Event 10079 may be used to prove that both the SQL statement SET ROLE role_name IDENTIFIED BY password as well as DBMS_SESSION.SET_ROLE send the role’s password unencrypted to the DBMS server. This means that any user who knows enough about Oracle Net, can get the unencrypted role password from a packet dump. Since an end user cannot add an ALTER SESSION statement to an application, an alternative way to dump Oracle Net packets is needed. All that is necessary is to copy tnsnames.ora and sqlnet.ora to the user’s home directory and to set TNS_ADMIN to the same directory. Then, after adding the following two lines to sqlnet.ora: