Oracle Database Security Considerations

By Kostas G. Gavrielidis

This Knowledge Brief presents important configuration and environment settings for Database Security considerations that would require implementation in their production environments. It proposes important areas of focus for HP engineers who engage in providing proactive consulting and remedial support in the Enterprise Database Security space on Oracle8i and Oracle9i UNIX and Windows production configurations.

## Methodology

We view Security in the Enterprise as an Enterprise-wide function bound by the Enterprise’s *IT Strategy* and *Operations* guidelines and affecting everything from *Enterprise Applications* to *Hardware Platforms*. Figure 1 illustrates the Security function along with the stack of dependencies that support the Enterprise Application.

We recommend that all database production configuration environments be monitored both proactively and reactively for security purposes as described in the following sections.

### Proactive Monitoring

* Identify and install patches or updates for all known database security vulnerabilities.
* Identify all database privileged users and review their access requirements. Limit each user’s ability to intentionally or accidentally harm the production databases.
* Ensure that database account passwords are changed from their defaults and review all privileged accounts.
* Verify that database files, configuration parameters, roles, etc. have the correct settings to allow adequate protection and auditing.
* Proactive monitoring should alert the Database Administrator (DBA) whenever a predefined security threshold has been reached or exceeded, while ensuring that the alert is created in sufficient time to remedy the problem before it becomes an outage.

### Reactive Monitoring

This activity includes the measurement of long-term performance and security trends to ensure that adequate lead-time is provided for hardware and software upgrades. The DBA must be alerted early enough to prevent any database resource or security risk that could result in a production outage.

## Enterprise Applications Security Considerations

The overall environment of an enterprise application and its related dependencies on the enterprise are represented in Figure 1. Enterprise wide functions, such as IT Strategy, Security and Enterprise Management, are necessary to support the enterprise application across the stack of dependencies. Also illustrated are the client services, persistent storage managers, transaction services, operating systems and network and hardware platforms. These are critical components that the enterprise application depends on or is built on.



Figure 1 - Enterprise Components in support of an EA

Our focus is primarily on security aspects and considerations for the persistence storage managers and the data they hold, as required by enterprise applications such as SAP R/3, Exchange, and the ERP and CRM implementations of the various vendors. The persistence storage managers we focus on here are primarily products from the Oracle Corporation, which run on UNIX and Windows platforms. Other persistent storage managers such Rdb, SQL Server, Ingres, SYBASE etc. may also have similar risks associated with security.

## Database Security Risks

The [Oracle9i Security Overview](http://www.stanford.edu/dept/itss/docs/oracle/9i/network.920/a96582.pdf) document lists the following database security risks and areas for potential attacks:

* Data tampering
* Eavesdropping and data theft
* Falsifying User Identities
* Password related threats
* Unauthorized access to database data
* Lack of accountability
* Complex user management requirements

In the same document, Oracle enumerates the different problem areas, the solutions available, identifies security technology areas and gives a list of Oracle products and features that address these security problems.

## Database Security Threats

Database security threats tend to come from two areas:

* External to the database. External threats to the database come from the operating system, the network, privileged users, and anything else that could cause damage or harm to the database configuration or its data.
* Direct or internal to the database. Internal threats are only possible from privileged users of the database.

### Operating System Level Threats to the Database

The following describes some of the most common external threats to the database and what you can do to help reduce them:

* Direct access to system privilege accounts such as root on UNIX and Administrator on Windows should not be widely available or well known to users.
* Don’t enable the remote shell/login.
* Direct access to the oracle user account should not be permitted
* Don’t run scripts and batch jobs as oracle or as another privileged user
* Only sudo[[1]](#footnote-1) to root should be allowed on UNIX production servers
* Only sudo to oracle accounts should be allowed on UNIX. DBA’s should log in to the server as themselves, and should not access the oracle account directly
* Use ssh[[2]](#footnote-2) for shell access, rather than telnet or rsh to provide remote shell access to the Oracle database server.
* The SYS and SYSTEM Oracle accounts should only be used for recovery and instantiation of the database.
* Grant limited DBA privileges to developers via the use of roles.
* Don’t use the schema owner account for routine maintenance.
* DBA’s should login as themselves WITH DBA privileges for maintenance.
* DBA access should be revoked from all individual accounts on production databases that are not in the production DBA group.

### Database and Data Level Threats to the Database

Security threats and related problems with any database are bound to occur, for example, when:

* Direct access to SYSTEM and SYS Oracle accounts is available and is not secured
* Direct access to all schema owners is permitted and is unsecured
* DBA’s login to the server and database as ORACLE and as SYSTEM for routine maintenance operations
* Developers have full access rights to production servers
* General db users and db support personnel have full privileges

## What can be done to reduce Database Security Threats?

Some of the steps the DBA can take in order to reduce the database security threats include the following:

* Identify and record all the software versions and patch levels on the production systems. Patch levels and versions can be found by checking the V$VERSION view. Also, the script products.sh[[3]](#footnote-3) can be used for Oracle versions up to version 8.
* Install only the features that are needed. When installing choose the custom option and de-select the options you do not need.
* Stop and disable all unnecessary services and daemons on the Oracle database server that are not needed or that Oracle does not use.
* Work on the definition and implementation of best practices with regards to database security.

## What can be done to reduce database security threats from the O/S level?

Some of the steps the DBA can take in order to reduce the database security threats at the operating system level include the following:

* The o/s user account chosen as the owner of the Oracle software should be the account that owns all of the files in the $ORACLE\_HOME/bin directory.

Lock and unlock the Oracle o/s account with the command:

$ /usr/sbin/passmgmt <username>

* Correct the Oracle file permissions
* Ensure that the permissions on $ORACLE\_HOME/bin are set to 0751 or less (set to 0755 in Oracle9iR2).

### Oracle on UNIX:

Check the setting of umask[[4]](#footnote-4):

 $ umask

Set the umask value to 022.

 $ umask 022

Set the owner, group and modes of the Oracle files with:

 $ chgrp -R oinstall $ORACLE\_HOME

 $ chown -R oracle $ORACLE\_HOME

### Oracle on Windows:

* Use the NTFS file system since FAT/FAT32 has no file system security.
* Create a user to own the Oracle software (do not call it oracle).
* A local user, not a domain user, should be used to install Oracle.
* Create an o/s group to be the OSDBA group (do not call it ORA\_DBA) and make it part of the admin group.
* Stop all other admin from accessing files owned by the Oracle software owner.
* Remove and do not allow the everyone group access to the ORACLE\_BASE, ORACLE\_HOME and to Oracle datafiles directory trees.
* Ensure that Windows file permissions are set to inherit the permissions of their top level directory.
* For the installation, use a temporary directory that is locked down. The /tmp directory is insecure because anyone can write to it. On UNIX use the /u01/tmp filesystem instead.
* Review membership of the OSDBA group.

On UNIX: $ grep -i dba /etc/group

On Windows: Review membership of the ORA\_DBA group

* Ensure the oracle account is not a member of the root group:

Review the results of the grep command:

$ grep -i root /etc/group

* Do not use the name DBA for the owner of the Oracle software.
* Check the permissions of the Oracle datafiles on UNIX:

$ ps -ef | grep pmon

$ cd <datafile directory>

$ ls -al

Change read/write only for owner:

$ chmod -R 600 \*

Change read/write for owner and group:

$ chmod -R 660 \*

* Hide the username and password from the ps[[5]](#footnote-5) command using the /nolog option on the sqlplus command as follows:

$ sqlplus /nolog <username>/<password>

## What can be done to reduce database security threats at the DB level?

Use the following to reduce database security threats at the Database level:

* Oracle parameter settings
* User passwords and privileges control
* Batch processing
* iSQL\*Plus
* Application Server and the Middle Tier

### Oracle Parameter Settings

* Set the Oracle parameter, DBLINK\_ENCRYPT\_LOGIN to TRUE in the init.ora file. In Oracle V7.1 and lower, the passwords could not be encrypted. On Oracle V7.3.4 and later, this parameter is set to FLASE by default. On Oracle 9iR2 the setting is unnecessary since all entries are encrypted.
* Set the ORA\_ENCRYPT\_LOGIN environment variable to TRUE on clients so they do not transmit the password in clear text.
* If access is allowed via the Java thin driver (OCI JDBC) then ensure that minimum privileges are granted to those users. The thin JDBC driver uses Java sockets to connect to the database, in order to implement a two-task communication protocol (TTC), and transmits clear text.
* Restrict users from compiling native PL/SQL code. Check the V$PARAMETER view for the Oracle PLSQL\_\* parameters. Check and remove the Oracle PLSQL\_\* parameters from the initialization file.
* The Oracle parameter O7\_DICTIONARY\_ACCESSIBILITY must be set to FALSE.

Setting this to FALSE stops those users or roles granted SELECT ANY TABLE access to the data dictionary. Only the SYS owner should have access to the data dictionary. In Oracle 7, 8 and 8i this parameter is set to TRUE by default. In Oracle9i, this parameter is set to FALSE by default.

* The Oracle parameter, REMOTE\_OS\_ROLES must be set and you should verify that it is set to FALSE. If not, any user can spoof the o/s and connect as any user.
* Ensure that the Oracle hidden parameter, \_SYSTEM\_TRIG\_ENABLED is set to TRUE. Otherwise none of the system triggers, in the database, will fire. System triggers can be used to monitor errors and security violations or to aid auditing so it is important that they operate properly.
* Ensure that the Oracle initialization parameter, AUDIT\_SYS\_OPERATIONS is set to TRUE. This is a new parameter in Oracle9iR2 and ensures that all actions by the SYS user are audited. This parameter can be set in the init.ora file or in the new binary parameter file: spfile, if used.

Check:

SQL> connect sys/<password>@<dbname> as sysdba

SQL> select name, value

from v$parameter

where name='audit\_sys\_operations';

Set:

 SQL> alter system

set audit\_sys\_operations=true scope=spfile;

 SQL> startup force;

Verify:

SQL> select name, value

 from v$parameter

 where name='audit\_sys\_operations';

### User passwords and privileges control

Some of the measures that can be taken to prevent issues with user passwords and elevated privileges include the following:

* Ensure that no database user has the ALTER SYSTEM and ALTER SESSION privileges.

Check to see who has these privileges with the SQL query:

 SQL> select grantee, privilege

 from dba\_sys\_privs

 where privilege in ('ALTER SYSTEM', 'ALTER SESSION');

Review all users or roles that have these privileges and remove them if necessary using the command:

 SQL> revoke alter system from <grantee>;

* If an export is performed against the production database with the intent of populating a test or development database, this can lead to security issues with the production passwords and with the database link passwords. Identify and lock all accounts with default passwords.

Identify: Numerous tools exist on the web to help with default password auditing. A good one is [here](http://www.petefinnigan.com/default/default_password_checker.htm).

Lock:

 SQL> connect sys/<password> as sysdba

 SQL> alter user dbsnmp account lock;

Verify Lock:

 SQL> connect dbsnmp/<password>

 ERROR:

 ORA-28000: the account is locked

Drop any accounts that are not required with:

 SQL> drop user pete cascade;

Change passwords with:

 SQL> alter user mike identified by mikepwd

 or

 SQL> connect sys/<password> as sysdba

 SQL> password <newpassword>

* Ensure the SYS and SYSTEM account passwords are changed from the defaults of CHANGE\_ON\_INSTALL and MANAGER and are changed every month. Passwords should contain at least twelve characters password and include two numeric characters and two punctuation characters.
* Default roles are similar to default users and have supplied default passwords.

For example:

 Role Name Default Role Password

 ---------- ----------------------

 ORD\_SERVER ODS

 WKADMIN WKADMIN

 WKUSER WKUSER

Query the view DBA\_ROLE\_PRIVS for a list of users using the command:

 SQL> select \* from dba\_role\_privs;

Try the role/passwords using the command:

 SQL> set role <rolename> identified by <rolepassword>;

* Check for all users who have the DBA privilege using:

 SQL> select grantee, default\_role, admin\_option

 from dba\_role\_privs

 where granted\_role = 'DBA';

Revoke the DBA role from users who don’t need it using:

 SQL> revoke dba from <user>;

* Additionally, the DBA role can be granted to the database administrators as a non default role. To use this escalated privilege, the administrator needs to issue the set role command:

 SQL> set role dba

this will be recorded in the audit trail if AUDIT ROLE has been enabled.

* Remove PUBLIC execute privilege from the following PL/SQL packages: utl\_file, utl\_tcp, utl\_http utl\_smtp, dbms\_random and dbms\_lob using the SQL revoke commands:

 SQL> revoke all on utl\_file from public;

- the utl\_file package can be used to access o/s file systems

 SQL> revoke all on utl\_tcp from public;

- the utl\_tcp package can be used to read and write sockets

 SQL> revoke all on utl\_http from public;

- the utl\_http package can be used to write content on a web browser

 SQL> revoke all on utl\_smtp from public;

- the utl\_smtp package can be used to send mail messages from the db server

 SQL> revoke execute on dbms\_random from public;

- the dbms\_random package can be used to generate random numbers and it is used in security features of applications

 SQL> revoke execute on dbms\_lob from public;

- the dbms\_lob package is used to access Large Objects (LOB's) and Character LOB’s (CLOB's). LOB’s can store up to 4GB of data in raw, binary or textformat

 SQL> revoke execute on owa\_util from public;

- the owa\_util package can be used from the web to show PL/SQL source and also to call arbitrary SQL.

* Set the DEFAULT and TEMPORARY TABLESPACE of all users to something other than SYSTEM TABLESPACE.

Check who has DEFAULT and TEMPORARY set to SYSTEM using:

 SQL> select username,

 default\_table\_space,

 temporary\_tablespace

 from dba\_users;

Make the change for the users. For example, for the dbsnmp user:

 SQL> alter user dbsnmp

 default\_table\_space tools

 temporary\_tablespace temp;

* Consider using tablespace quotas to avoid Denial of Service (DoS) attacks from users who have UNLIMITED TABLESPACE.
* Ensure the Oracle listener password has been set. In the $ORACLE\_HOME/network/admin/listener.ora file add:

 PASSWORDS\_{<Listener\_Name>}=(<Password1>[,<Password2>])

Or use the LSNRCTL program:

 # lsnrctl

 LSNRCTL> set current\_listener <listener\_name>

 LSNRCTL> change\_password

 Old password:

 New password:

 LSNRCTL> save\_config

* Ensure the Oracle $ORACLE\_HOME/network/admin/listener.ora parameter ADMIN\_RESTRICTIONS\_{<Listener\_name>} is added and set to ON (for example: ADMIN\_RESTRICTIONS\_{<Listener\_name>}=ON )

This parameter prevents the listener from accepting SET commands either locally or remotely while it is running. This parameter was added to prevent reading and writing of O/S files when the listener password is not set.

* Check database links SYS.LINK$ view for clear text password:

 SQL> select name, host, password, authusr, authpwd, userid

 from sys.link$

 where password is not null;

If the db applications must have public database links, then restrict access to the SYS.LINK$ view; otherwise, use private database links.

* Ideally there should not be direct developer access to the production database. Check for all developer accounts in production. Place production databases on a different network segment from the test and development databases.

### Batch Processing

* All batch processes should ideally access the database through one account designed specifically for this purpose.
* Do not use external accounts for batch processes. On the surface, they seem like a good choice, but they provide a simple way to access the database.
* If a scheduler is used, make sure the process to obtain and use passwords is secure. Ideally, passwords should not be a command line parameter or an environment parameter.
* Use a time based approach to enable accounts that can access batch processes and conversely, disable all other accounts while batch processes run. Having only one account available during batch processing makes it harder for an attacker to gain access, and also makes attempts to access the database easier to spot.

### iSQL\*Plus[[6]](#footnote-6)

* Disable iSQL\*Plus if it is not needed; this tool should not be enabled on a production server.

To disable iSQL\*Plus, do the following:

* 1. Stop the Oracle HTTP Server
	2. Find the oracle\_apache.conf file. This should be located as follows:

Windows: %ORACLE\_HOME%/Apache/Apache/conf

UNIX: $ORACLE\_HOME/Apache/Apache/conf

* 1. Comment out the line, including isqlplus.conf by adding the comment character “#”.

 Example on UNIX:

 # include "ORACLE\_HOME/sqlplus/admin/isqlplus.conf"

 where ORACLE\_HOME is the path to the Oracle home directory.

* 1. Save the file and restart the Oracle HTTP Server. iSQL\*PLus should no longer be available.
* To avoid the DBA URL being automatically authenticated, start the Oracle HTTP Server (OHS) as an operating system user who is not a member of the ORSDBA or OSOPER groups on UNIX, or who is not a member of the ORA\_DBA, ORA\_<SID>\_DBA, ORA\_OPER or ORA\_<SID>\_OPER on Windows.

Starting the OHS as a user who is not in these groups will stop the operating system authentication that allows the privileged connections "as sysdba" and "as sysoper". Using the privileges field in the login screen to access as SYSDBA or SYSOPER will still work when using the OHS authentication file.

## Summary

In this Knowledge Brief, we presented and summarized the important configuration and environment characteristics related to Oracle Database security on UNIX and Windows HP production environments. We also looked at different threats to the database and how these could be prevented.

## References

A Security Checklist for Oracle9i - <http://www.oracle.com/technology/deploy/security/oracle9i/pdf/9i_checklist.pdf>

Database Encryption in Oracle9i - <http://www.oracle.com/technology/deploy/security/oracle9i/pdf/f5crypt.pdf>

Oracle Security Handbook by Marlene Theriaulf and Arch Newman, from Oracle Press

Oracle Security: Step-by-Step by Pete Finnigan, SANS Press

Oracle Security site - <http://otn.oracle.com/deploy/security>

Oracle9i Security Overview - <http://www.stanford.edu/dept/itss/docs/oracle/9i/network.920/a96582.pdf>

Oracle9i Database Security for eBusiness - <http://www.oracle.com/technology/deploy/security/oracle9i/pdf/9isecbpa.pdf>

PeteFinnigan.com Limited - <http://www.petefinnigan.com/whatsnew.htm>

Secure Configuration Guide for Oracle9i R2 - <http://www.oracle.com/technology/deploy/security/oracle9i/pdf/9ir2_checklist.pdf>

Sudo - <http://www.courtesan.com/sudo>

1. sudo (superuser do) allows a system administrator to give certain users (or groups of users) the ability to run some (or all) commands as root while logging the commands and their arguments. [↑](#footnote-ref-1)
2. ssh (secure shell) is a program for logging into a remote system and for executing commands on the remote system. It is intended to replace rlogin and rsh, and provide secure encrypted communications between two un-trusted hosts over an insecure network. [↑](#footnote-ref-2)
3. The script products.sh is available in the Oracle Scripts book by Brian Lomasky and David C. Kreines, an O’Reilly & Associates publisher. [↑](#footnote-ref-3)
4. The umask command lists or changes the default permissions assigned by the system whenever a new file or directory is created. The number specified as a parameter to the umask command works in an opposite manner to the number given to the chmod command. The mask serves to remove permissions as opposed to granting them. That is, the digits in the umask number are 'subtracted' from 777 for directories and from 666 for files when we are creating their initial permissions. In our case after the “$ umask 022” command, if we create new files their default permissions will be 644 (=666 - 022, i.e. -rw-r--r--), and when we create new directories their default permissions will be 755 (=777 - 022, i.e. drwxr-xr-x). [↑](#footnote-ref-4)
5. The ps (process status) command shows the currently running processes on UNIX systems. [↑](#footnote-ref-5)
6. iSQL\*Plus is new in Oracle9i R2 (9.2). iSQL\*Plus generates output in the form of HTML tables, has a command history tool and an option for privileged users to connect as SYSDBA or SYSOPER via a HTTPD password file.

The URL for iSQL\*Plus on a local system by default is: <http://localhost/isqlplus> [↑](#footnote-ref-6)