SQL-Retriever Security

Successive ODBC standards and the solutions that are built around it, are often criticized for lack of security (which is itself a function of ODBC's openness).

Once the solution is deployed, SQL-Retriever's **Security Manager** feature is used by system administrators to limit access to data. It is a UNIX-based program that allows or denies read and write access from individuals or particular Windows applications to databases or particular tables within them. It does not use or require any of the database-specific management tools. System administrators configure rights by defining access privileges in a simple configuration file supplied with SQL-Retriever. They can configure groups of users, perhaps representing departments, and apply restrictions to the groups. Similarly, suites of applications can be configured, so that users in certain departments will be able to use only the programs they need. This avoids the risk of people – unwittingly or wilfully – damaging the UNIX data with undesired functionality available in other off-the-shelf products.

Security Manager is discrete from the proprietary database vendor security mechanisms; it's control operates over all UNIX SQL databases involved in the solution.

In addition to the Security Manager's control of access to data, 'live' security concerns are addressed by additional communications security inherent in SQL-Retriever. This option encrypts the user name and password before it is sent across the network.

Security Manager in detail

Security Manager is a component of the host module of the SCO Vision ODBC driver that allows a System Administrator to grant or revoke users access to the database or to specific tables in the database. Security Manager allows access to the database to be granted or revoked by application, enabling restrictions coded into the application itself (for example, if the application will only allow the user to perform queries) to complement restrictions imposed by Security Manager. Security Manager is started when the UNIX server starts by specifying the -r option in the entry for the server in the servers database.

Granting and revoking privileges

Entries in Security Manager's configuration file **sqlrsec.conf** determine what privileges a user has. A sample configuration file **sqlrsec.sample** is provided with example entries illustrating the use of Security Manager to grant and restrict access to tables in the demonstration database. You should save the file as **sqlrsec.conf** and then modify the example entries or create new ones according to your own requirements using a text editor (you need Superuser privilege to do this).

A number of keywords can be used to create groups of users, databases, applications, SQL statements and so on that may be used in entries granting and revoking privileges. The example below demonstrates the use of **USERGROUP** to create the SALES group of which andy and lee are members.

```
USERGROUP SALES = andy, lee
```

The format of entries granting and revoking privileges is:

GRANT | REVOKE *user database:reserved:owner:table:application* = *privilege,...*

Note that the fields in these entries do not have to be groups or members of groups defined in **sqlrsec.conf**.

The following entry grants the members of the SALES group the ability to perform queries on all tables in Scodemo using SQLGold.

```
GRANT SALES scodemo:ALL:ALL:USCOMMERCETAB:gold32 = SELECT
```

Alternatively, you may choose to limit the privileges a user has on a database by specifying an application with built-in restrictions. For example, in the following entry the Visual Basic demonstrator application VB4 Demo is specified in the *application* field. As VB4 Demo will only allow a user to execute SELECT statements, the effect of the entry below is the same as the entry in the preceding example:

```
GRANT SALES scodemo:ALL:ALL:USCOMMERCETAB:vb4demo = ALL
```

If an application launches another application to query a database (for example, one of the ways Excel works with external data is to use Query) then both need to be specified either in separate **GRANT** entries or in one **GRANT** entry where a group defined to contain both applications is specified.

Privileges have first to be granted before they can be revoked. In the example below the ability of lee (a member of the SALES group) to perform queries on the Sales table is revoked.

```
REVOKE lee scodemo:ALL:ALL:sales:gold32 = SELECT
```

ALL or the asterisk character (*) can be used as a wildcard in any field. If **ALL** is specified in the *privilege* field then a predefined group of SQL statements will be used (refer to the comments in **sqlrsec.conf** for a list). In situations therefore where it is desirable to grant all users all privileges apart from a few specific exceptions you may choose to use the following entry:

```
GRANT ALL ALL:ALL:ALL:ALL:ALL = ALL
```

and then selectively revoke the privileges of appropriate users.

The order in which entries granting and revoking privileges are read by Security Manager is not necessarily the physical order in which they appear in the configuration file. Security Manager uses the *user* field to sort the entries, processing entries where **ALL** or * is specified in the field first, where a group of users is specified second, and where an individual user is specified last. If multiple entries have the same value in the user field, then the *database* and *application* fields are used to differentiate between them, again with entries containing the most generic values in these fields being processed first and the most specific last. In the following example the **GRANT ALL** entry will be read first, the **REVOKE ALL** entry second (because even though **ALL** is specified in the *user* field of both these entries, a specific database is specified in *database* field of the **REVOKE** entry) and the remaining **GRANT** entry last (because a specific user is specified in the *user* field).

```
GRANT ALL ALL:ALL:ALL:ALL:ALL = CREATE TABLE, CREATE INDEX, SELECT

GRANT timp scodemo:ALL:ALL:ALL:ALL = DELETE

REVOKE ALL scodemo:ALL:ALL:ALL:ALL = SELECT
```

The sqlrsec.conf file

The following is a transcript of the sqlrsec.conf file delivered with the SQL-Retriever product.

```
# Sample Security Manager configuration file. You should save the file as
# sqlrsec.conf and then modify the example entries or create new ones
# according to your own requirements using a text editor (you need
# Superuser privilege to do this).
#
# The following keywords can be used to create groups of users,
# databases, applications, SQL statements and so on for use in entries
# granting and revoking privileges:
#
# USERGROUP DBGROUP TABLEGROUP APPGROUP SQLGROUP
#
# The format of entries used to define groups is:
#
# <keyword> <groupname> = <groupmembers,...>
#
# For example,
#
# SQLGROUP DDL = ALTER TABLE, CREATE VIEW, CREATE TABLE, DROP TABLE
#
# uses the SQLGROUP keyword to create a group called DDL containing a
# number of SQL statements. Separate multiple group members with commas.
#
# Use the GRANT and REVOKE keywords to grant and revoke privileges.
# Format of entries granting and revoking privileges is:
#
# GRANT | REVOKE <user> <database>:<reserved>:<owner>:<table>:\
# <application> = <privilege,...>
#
# where <user> is the UNIX username of the individual user or
# group of users to whom privileges are being granted or revoked,
# <database> is the database or group of databases, <owner> is the owner
# of the database tables, <table> is the table or group of tables,
# <application> is the application's executable file name minus the
# extension or group of applications, and <privilege> is the SQL statement
# or group of SQL statements to be granted or revoked. Separate multiple
# privileges with commas. Note that fields in these entries do not have to
# be groups or members of groups defined in sqlrsec.conf. ALL or the
# asterisk character (*) can be used as a wildcard in any field. If ALL is
# specified in the privilege field then the following predefined group of
# SQL statements will be used:
#
# SELECT, INSERT, UPDATE, DELETE, GRANT, REVOKE, CREATE TABLE, CREATE INDEX,
# CREATE VIEW, DROP TABLE, DROP VIEW, DROP INDEX, EXECUTE PROCEDURE,
# ALTER TABLE
#
# The following sample entries are provided to illustrate the use of
# Security Manager to grant and restrict access to tables in the Vison
# Family demonstration database Scodemo. Refer to the Reference manual
# for your Vision Family product for information if you have not yet
# created the demonstration database. Substitute the names of the users in
# the example with the names of members of your organization.

USERGROUP SALES = andy, lee
USERGROUP ACCOUNTS = chris, sarah
USERGROUP SYSADMIN = bob

DBGROUP USCOMMERCEDB = scodemo
```

```
TABLEGROUP USCOMMERCETAB = customer, orders, items, stock, manufact, sales, nominal

APPGROUP SALESAPP = msaccess, gold32
APPGROUP ACCOUNTSAPP = msaccess, excel, msqry32
APPGROUP SYSADMINAPP = msaccess, gold32

SQLGROUP SYSADMINSQL = INSERT, UPDATE, DELETE, GRANT, REVOKE, CREATE TABLE, CREATE INDEX,
CREATE VIEW, DROP TABLE, DROP VIEW, DROP INDEX, EXECUTE PROCEDURE, ALTER TABLE
SQLGROUP DDL = ALTER TABLE, CREATE VIEW, CREATE TABLE, DROP TABLE
SQLGROUP DML = SELECT, INSERT,  DELETE

# The following entry grants the members of the SALES group the ability
# to perform queries on all tables in Scodemo using SQLGold.
#
GRANT SALES scodemo:ALL:ALL:USCOMMERCETAB:gold32 = SELECT
#
# In the following entry the Visual Basic demonstrator application VB4 Demo
# is specified in the <application> field. As VB4 Demo will only allow a
# user to execute SELECT statements, the effect of the entry below is the
# same as the entry below.
#
GRANT SALES scodemo:ALL:ALL:USCOMMERCETAB:vb4demo = ALL
#
# Although lee is a member of the SALES group, he is not a manager and
# therefore his ability to perform queries on the Sales table (containing
# information about the performance of his colleagues) is revoked in the
# entry below. Note that a privilege has first to be granted before it can
# be revoked.
#
REVOKE lee scodemo:ALL:ALL:sales:gold32 = SELECT
#
# As the nominal table is a general ledger for accounts it is of no direct
# interest to members of the SALES group and their ability to query this
# table is revoked.
#
REVOKE SALES scodemo:ALL:ALL:nominal:gold32 = SELECT
#
# In the following entry, members of the group SYSADMIN (System Administrators)
# are granted the ability to use the SQL statements defined in the group
# SYSADMINSQL. SYSADMINSQL contains many SQL statements including CREATE
# TABLE and ALTER TABLE, however SELECT is omitted, therefore although
# SYSADMIN members can alter the structure of the database they aren't
# allowed to query the (potentially sensitive) data.
#
GRANT SYSADMIN scodemo:ALL:ALL:USCOMMERCETAB:gold32 = SYSADMINSQL
#
# In situations where it is desirable to grant all users all privileges
# apart from a few specific exceptions you may choose to use the following
# entry:
#
# GRANT ALL ALL:ALL:ALL:ALL:ALL = ALL
#
# and then selectively revoke the privileges of appropriate users. For
# example, in the entry below the SQL statements contained in SYSADMINSQL
# are revoked. As SYSADMINSQL doesn't contain the SELECT statement, the
# right of ACCOUNTS members to perform queries remains.
#
```

```
# REVOKE ACCOUNTS scodemo:ALL:ALL:USCOMMERCETAB:ACCOUNTSAPP = SYSADMINSQL
#
# In the following entry, members of ACCOUNTS are prohibited from
# performing queries on the Sales table.
#
# REVOKE ACCOUNTS scodemo:ALL:ALL:sales:ACCOUNTSAPP = SELECT
#
# The previous two entries specified the ACCOUNTSAPP group of applications
# (containing Access, Excel, and Query) in the <application> field. If a
# member of the ACCOUNTS group was to use another application for example
# SQLGold, he wouldn't be subject to the restrictions imposed by the two
# REVOKE entries and would have all privileges in-tact by virtue of the
# initial GRANT entry. This is addressed in the following entry. It is the
# responsibility of the System Administrator to be aware of what
# applications are in use at the site and construct appropriate entries in
# the configuration file accordingly.
#
# REVOKE ACCOUNTS scodemo:ALL:ALL:USCOMMERCETAB:gold32 = ALL
```

Location

$VISION_ETCDIR/sqlrsec.conf

Description

Note The sample file **$VISION_ETCDIR/sqlrsec.sample** provided to illustrate the use of Security Manager with the demonstration database should be saved as **sqlrsec.conf** and then modified to your own requirements. You need Superuser (root) privilege to do this.

sqlrsec.conf is only available if SQL-Retriever is installed. It is the configuration file for Security Manager. It is an ASCII text file that can be changed using a text editor providing you have Superuser (root) privilege. Do not change the access permissions of this file. The file contains entries used to grant and revoke privileges and definitions of groups of users, databases, applications, SQL statements and so on that may be used in these entries. The format of entries used to define groups is:

keyword groupname = *groupmember*,...

where *keyword* is the appropriate keyword for the group to be created, as shown below.

Keyword	Groups
USERGROUP	UNIX usernames of users or groups of users
DBGROUP	Databases or groups of databases
TABLEGROUP	Database tables or groups of tables
APPGROUP	Applications' executable file name minus the extension or groups of applications
SQLGROUP	SQL statements or groups of SQL statements

Separate multiple *groupmembers* with commas. For example:

```
APPGROUP ALLAPPS = msaccess, excel, msqry32, SALESAPPS
```

uses **APPGROUP** to create a group of applications named ALLAPPS containing Microsoft Access, Excel, Query, and the applications defined in the group SALESAPPS.

Format of entries granting and revoking privileges is:

GRANT | REVOKE *user database:reserved:owner:table:application* = *privilege,...*

where *user* is the UNIX username of the individual user or members of the group of users to whom privileges are being granted or revoked, *database* is the database or group of databases, *owner* is the owner of the database tables, *table* is the table or group of tables, *application* is the application's executable file name minus the extension or group of applications, and *privilege* is the SQL statement or group of SQL statements to be granted or revoked. Separate multiple privileges with commas. Note that fields in these entries do not have to be groups or members of groups defined in **sqlrsec.conf**.

ALL or the asterisk character (*) can be used as a wildcard in any field. If **ALL** is specified in the *privilege* field then a predefined group of SQL statements will be used (refer to the comments in **sqlrsec.conf** for a list).

Example

The following example grants members of the SALES group the ability to perform queries on all tables in the demonstration database. This privilege is then revoked on certain tables.

```
GRANT SALES scodemo:ALL:ALL:USCOMMERCETAB:gold32 = SELECT
REVOKE lee scodemo:ALL:ALL:sales:gold32 = SELECT
REVOKE SALES scodemo:ALL:ALL:nominal:gold32 = SELECT
```