Using ISQL to View the Cisco Secure Database

Document ID: 13840

Introduction

The default database software that comes with Cisco Secure UNIX 2.0 and later is SQLAnywhere. (Oracle or Sybase can be purchased in lieu of the default.) This document explains the database structure, gives an example of records, illustrates typical queries, and shows how to execute the queries through the command line interface (CLI) (ExecSql) or the SQLAnywhere GUI (isql). This document also discusses the ViewProfile and DBClient.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on Cisco Secure UNIX 2.0 and later.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Database Structure

The database structure is described in the Cisco Secure ACS for UNIX documentation. Consult the *UNIX Reference Guide* chapter on *Cisco Secure ACS Database Structure*. The same database schema is used whether the database is SQLAnywhere, Oracle, or Sybase. The Cisco Secure GUI (or a command−line interface utility such as AddProfile, CSImport, CSMigrate, and so on) is used to add user and group information in an orderly fashion. This relational database consists of tables. The tables have records, and the records have fields. The field that inter−relates all table information on a particular user is the "profile_id" as illustrated by the Example of Records:

Example of Records

In this example:

- The username of the use ("user_name") is found in the "cs_user_profile" table with profile_id = 90.
- The group membership of the user ("member") is found in the "cs_profile" table with profile_id = 90.
- The complex information in the profile of the user such as privilege−level, commands, and allow/refuse statements ("blob_data") are found in the "cs_profile_blob" table with profile_id = 90.
- The password of the user ("pwd_value") & type ("pwd_type") are found in the "cs_password" table with profile_id = 90.
- The enable−15 password of the user is found in the "cs_privilege" table with profile_id = 90.
- Information on the group of the user ("e−group") is in the "cs_group_profile" table with profile_id = 72.

You see where bits and pieces of the user reside in six different tables. While your query of the database with ExecSql or isql is fairly simple, the manual modification of the database with ExecSql or isql is not recommended due to this complexity. The use of the GUI and CLI utilities, the tables and profile_ids in the database are kept in sync.

```
user = example{
profile_id = 90
profile_cycle = 1
member = e−group
password = clear "********"
privilege = clear "********" 15
service=shell {
set priv−lvl=15
cmd=show {
permit "run"
}
allow "^10.21.1.188" ".*" ".*"
refuse ".*" ".*" ".*"
}
}

group = e−group{
profile_id = 72
profile_cycle = 1
}
```

Typical Queries

Note: Complete SQLAnywhere help is available on the Cisco Secure UNIX system. Point a browser to http://ip_address_of_csunix_box/SQLHelp/dbeng50.htm

- Pull a list of all users:

```
select * from cs_user_profile
```

- Pull certain fields in all accounting records:

```
select user_name, start_time, stop_time from cs_user_accounting
```

- Pull certain accounting record fields by date and time:

```
select user_name, start_time, stop_time from cs_user_accounting
where start_time < "2001-05-10" and user_name = "joeuser"
```

- Pull pieces of the user/group profile from each table (our user had profile_id=90 and the group profile_id was 72):

    - `select * from cs_user_profile where profile_id = 90`
    - `select * from cs_profile where profile_id = 90`
    - `select * from cs_profile_blob where profile_id = 90`
    - `select * from cs_password where profile_id = 90`
    - `select * from cs_privilege where profile_id = 90`
    - `select * from cs_group_profile where profile_id = 72`

Command Line Interface (ExecSQL)

The ExecSql utility is in the $BASE/utils/bin directory ($BASE is the directory where Cisco Secure UNIX is installed).

```
cd $BASE/utils/bin
./ExecSql "your_select_statement_here"
```

The output can be scripted or redirected with typical UNIX commands. For example,

```
./ExecSql "your_select_statement_here" > myfile
```

SQLAnywhere GUI (ISQL)

Set Environment Variables

Provided Cisco Secure is up and running, very little is required to use isql.

1. Ensure these environment variables are set in your shell:

```
SQLANY=/csecure/SYBSsa50
  LD_LIBRARY_PATH=/csecure/SYBSsa50/lib
  SATMP=$SQLANY/tmp
```

   Note: In this example, **csecure** is a link to **$BASEDIR**. **$BASEDIR** is the full path to the Cisco Secure directory. Your path does not say **csecure** but should instead be the full path to **$BASEDIR**.
2. You must also have Cisco Secure in your path, as shown in this example.

```
PATH=/bin:/usr/bin:/usr/ucb:/etc:/csecure/SYBSsa50/bin:/usr/sbin:/sbin:.
```

3. In order to make sure these environment variables are correctly set, enter:

```
source $BASEDIR/utils/bin/env_setup
```

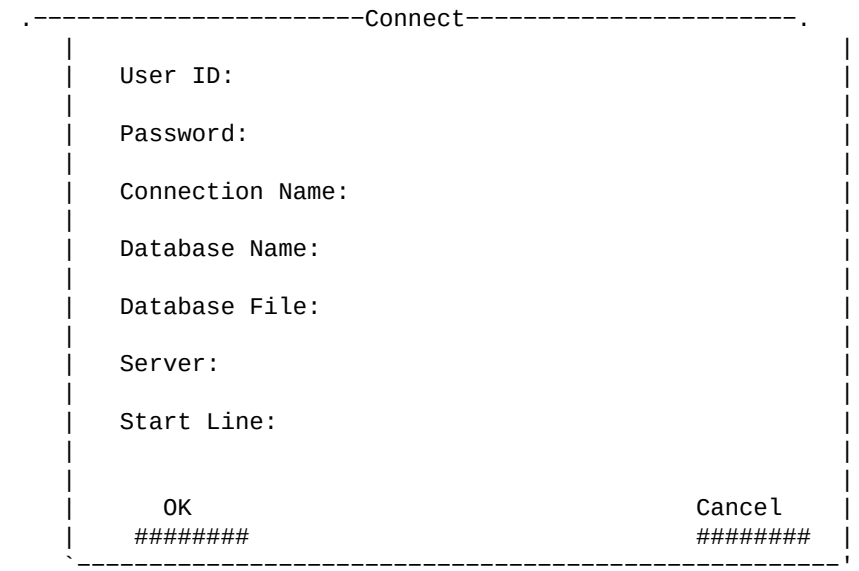
Access Database Information

Complete these steps:

1. At the command line enter:

```
isql
```

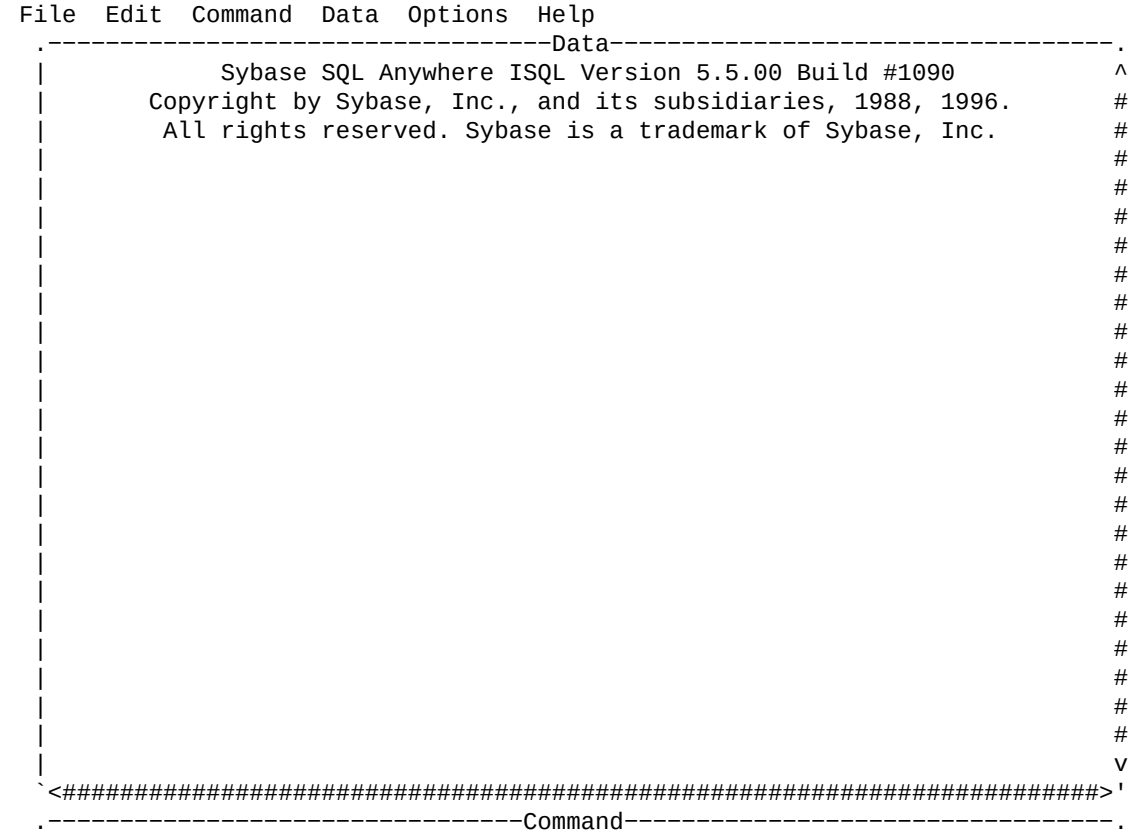
   A screen appears similar to this:



2. Enter only the Database Administrator name (DBA) and password (SQL). None of the other lines are required.

   Navigation Hint: In order to navigate, press the **Tab** key to get to the line you want.
3. Tab down to OK and press **Return**.

   You then see a screen similar to this:



Your cursor is located toward the bottom of the screen in the Command section.

4. With the cursor you can enter standard SQL statements in the Command section. Enter the statement and press the Execute key, **F9**.

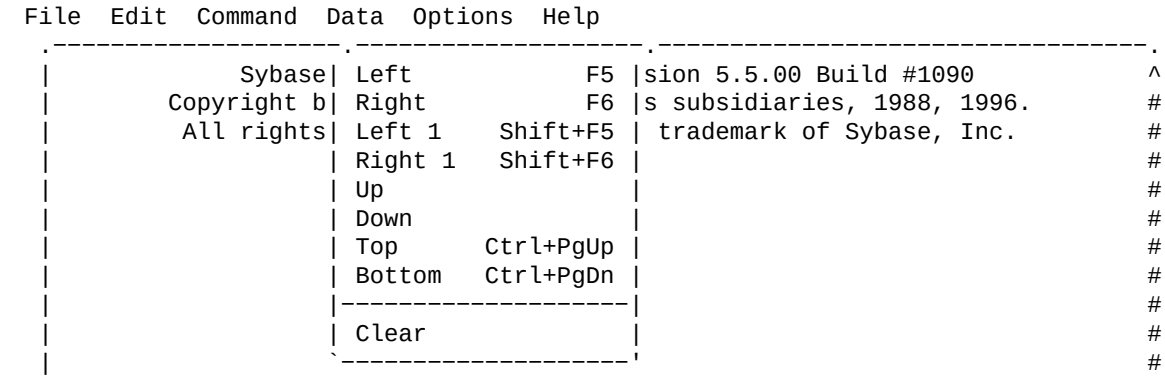
   A good example of a SQL statement to enter is:

   ```
   select * from cs_user_accounting
   ```

   When you press **F9** all the information held in the **cs_user_accounting** table displays in the Data area. Much of the data appears off the screen. In order to view it you need to scroll around. The easiest way to scroll the data display is to use the Data menu options.

   Navigation Hint: Press **Ctrl−A** to activate the menus. Then press **D**, the first letter in the Data menu. In order to access the other menus, enter **Ctrl−A**, then the first letter of the menu name.

5. Press **Ctrl−A D** to open the Data menu at the top of the display. You see something like this:



   You can tell from the menu that **F5** moves the display left and **F6** moves it to the right. You can use the keys or choose an option from the menu to scroll the data display.

   Hint: You can save the data in the display to a disk file. In order to save your output into a file, append ># and a filename to the end of the command you entered in the Command area.

   An example is:

   ```
   select * from cs_user_accounting ># /tmp/accountingstuff
   ```

   Press **F9** to find your output in **/tmp**.

   In summary, to get to the menu items press **Crtl−A** and the first letter of the menu item. Remember to press **F9** to execute your SQL statements.

Exiting ISQL

The easiest way to exit from isql is to press **Ctrl−A F** and choose **Exit** from the File menu.

ViewProfile

If the userid or group is known, these can be seen with these commands:

```
$BASE/CLI/ViewProfile −p 9900 −u name_of_user
$BASE/CLI/ViewProfile −p 9900 −g name_of_group
```

DBClient Utility

DBClient is a limited−functionality tool which can be invoked with this command:

```
$BASE/DBClient −p 9900
```

Note: This requires the same username/password as for web access.

```
# ./DBClient −p 9900
Username: superuser
Password:
Request Types:
create, delete, update, replace, get, lock, unlock, query,
    insert_accounting, get_accounting, admin_Commands, is_unlock, exit
Request type: get
Data(get): (to quit type: )
user=abcde (& hit enter twice!)
Requesting Command: get
Response:
Response Type:SUCCESS
Response Data Size: 78
Response Data:
user = abcde{
profile_id = 88
profile_cycle = 1
password = clear "********"
}
```

Valid "gets" other than per−user are:

- group=group_name
- user=* (all users)
- group=* (all groups)

NetPro Discussion Forums − Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums − Featured Conversations for Security
Security: Intrusion Detection [Systems]
Security: AAA
Security: General
Security: Firewalling

Related Information

- **Cisco Secure ACS for UNIX Support Page**
- **Documentation for Cisco Secure ACS for UNIX**
- **Technical Support & Documentation − Cisco Systems**

Updated: May 14, 2009 Document ID: 13840