

TSAR consists of the following components:

- **TSAR Server:** Running as a started task under z/OS UNIX System Services, the TSAR server makes the entire RACF database available to ISPF and Windows™ clients.
- **GUI Client:** Running on Windows and communicating through TCP/IP with the TSAR server, the GUI client provides a Java-based interface for RACF administration.
- **ISPF Client:** Running under TSO/E on z/OS—The ISPF client provides a set of ISPF panels that access the TSAR server.

The following key concepts are used within the TSAR client interface.

Database: A TSAR server includes a database that contains a mirror image of one RACF database and a database that stores configuration information for required clients. Each TSAR server supports one RACF database. One configuration database can support multiple TSAR servers (and thus multiple RACF databases). The configuration database can be stored on the same TSAR server as one of the mirror RACF databases, or it can be stored on a separate TSAR server.

The client database contains the TCP/IP and server information required to connect to a TSAR server. Each client has one database definition stored locally to acquire configuration data from the server.

Entry: An entry is an entity containing either RACF information or TSAR configuration data. All entries consist of a unique name and any number of attribute-value pairs. The permitted types of attributes and value formats are determined by the server schema that resides on the TSAR server.

Attribute: An attribute is the name of a field used by the TSAR server to store information as part of an entry. Each entry has one or more attributes associated with it. Attribute definitions, such as user ID, name, and data, are stored as entries in the configuration database, and can be maintained using the TSAR client GUI or ISPF interface.

View: A view (or interactive report) represents a particular query against the TSAR server. It consists of:

- Selection criteria, which are either defined within a default view, or specified using a search filter

- Display fields, which determine the fields included in the result set
- Sort fields and sort order, which control the display of the result set.

The result set of a view is displayed as a list of entries, which can be modified if the user is authorized. View definitions are stored as entries on the TSAR server, and can be customized, copied, and more using the client GUI or ISPF interface.

Tool: A tool performs a particular set of operations against the RACF database. It consists of:

- A program executed against the server database
- A view that displays the result set following execution of the program.

The results of a tool action are displayed as a list of entries, and tool definitions are stored as entries on the TSAR server.

TSAR server

The server provided with TSAR runs under z/OS UNIX System Services. One TSAR server is required for each RACF database that is to be maintained by TSAR. Both the Java-based Windows client and the ISPF client included with TSAR can interact with multiple TSAR servers, providing administrators the ability to manage multiple RACF databases from one client session.

The TSAR server maintains a synchronized image of the RACF database. The TSAR server consists of the following components:

- Database that mirrors the RACF database
- RACF to TSAR server (RACF2LDAP)
- TSAR server to RACF (LDAP2RACF)
- Configuration database.

Mirror database

The mirror database represents a real-time image of the RACF database as it resides on the host z/OS system. The RACF database and mirror database are automatically synchronized by the RACF2LDAP and LDAP2RACF processes.

How it works - RACF to TSAR Server (RACF2LDAP)

The RACF2LDAP process updates the mirror database to reflect the current status of the RACF database.

Whenever a change is made to the RACF database, RACF2LDAP intercepts the system message generated by the RACF command (SMF 80 record). The RACF command is then translated into an equivalent server modify command that updates the mirror database accordingly.

How it works: TSAR Server to RACF (LDAP2RACF)

The LDAP2RACF process modifies the RACF database to reflect changes initiated within TSAR (GUI client or ISPF). Whenever users make a change to the RACF database using TSAR, LDAP2RACF translates the server modify command into an equivalent RACF command to update the RACF database accordingly. When the change has been made to the RACF database, RACF2LDAP processes and reflects the change within the mirror database using the RACF2LDAP process above.

Configuration database

The configuration database acts as a central repository of configuration data for all components of the TSAR server.

Security Administration on the GUI client

When you have successfully connected to a server using the GUI client, a list of available RACF databases is displayed in what is defined as the TSAR Tree View screen. You must connect to and authenticate to a RACF database in order to view or modify its RACF data. The Tree View screen allows you to drill down through a database's entries, create or delete entries, view the field information for selected entries, and execute views and queries.

Summary

IBM Tivoli Security Administrator for RACF provides a flexible, easy-to-use interface to view and maintain all types of profiles in the RACF database. Whether the administrator uses traditional interfaces to RACF or the new interfaces provided by TSAR, the underlying security features and controls of RACF are maintained. So long live TSAR!

For more information on IBM Tivoli Security Administrator for RACF, visit: ibm.com/software/tivoli/products/security-admin-racf/.

