Chap (Database security & auditing)

User Authentication

To prevent unauthorized use of a database username, Oracle provides user validation via three different methods for normal database users:

- authentication by the operating system
- authentication by a network authentication service
- authentication by the associated Oracle database

For simplicity, one method is usually used to authenticate all users of a database. However, Oracle allows use of all methods within the same database instance.

Oracle also encrypts passwords during transmission to ensure the security of client/server authentication.

Because database administrators perform special database operations, Oracle requires special authentication procedures for database administrators.

Authenticating Users Using the Operating System

If your operating system permits, Oracle can use information maintained by the operating system to authenticate users. The benefits of operating system authentication are the following:

- Users can connect to Oracle more conveniently (without specifying a username or password). For example, a user can invoke SQL*Plus and skip the username and password prompts by entering

```
SQLPLUS /
```

- Control over user authorization is centralized in the operating system; Oracle need not store or manage user passwords. However, Oracle still maintains usernames in the database.
- Username entries in the database and operating system audit trails correspond.

If the operating system is used to authenticate database users, there are some special considerations with respect to distributed database environments and database links;

Authenticating Users Using Network Authentication

If network authentication services, such as DCE, Kerberos, or SESAME, are available to you, Oracle can accept authentication from the network service. To use a network authentication service with Oracle, you must also have the Oracle Secure Network Services product.

If you use a network authentication service, there are some special considerations for network roles and database links.

Authenticating Users Using the Oracle Database

Oracle can authenticate users attempting to connect to a database by using information stored in that database. You must use this method when the operating system cannot be used for database user validation.

When Oracle uses database authentication, you create each user with an associated password. A user provides the correct password when establishing a connection to prevent unauthorized use of the database. Oracle stores a user's password in the data dictionary. However, all passwords are stored in an encrypted format to maintain security for the user. A user can change his/her password at any time.

Password Encryption while Connecting

To better protect the confidentiality of your passwords, Oracle allows you to encrypt passwords during client/server and server/server connections. If you enable this functionality on the client and server machines, Oracle will encrypt passwords using a modified DES (Data Encryption Standards) algorithm before sending them across the network.

Database Administrator Authentication

Database administrators must often perform special operations such as shutting down or starting up a database. Because these operations should not be performed by normal database users, the database administrator usernames need a more secure authentication scheme. Oracle provides a few methods for authenticating database administrators.

Depending on whether you wish to administer your database locally on the same machine on which the database resides or if you wish to administer many different database machines from a single remote client, you can choose between operating system authentication or password files to authenticate database administrators

Introduction to Auditing

***Auditing* is the monitoring and recording of selected user database actions**.
Auditing is normally used to

Why Is Auditing Used?

You typically use auditing to perform the following activities:

- **Enable accountability for actions.** These include actions taken in a particular schema, table, or row, or affecting specific content.
- **Deter users from inappro**priate actions based on that accountability.
- Investigate suspicious activity. For example, if a user is deleting data from tables, then a security administrator might decide to audit all connections to the database and all successful and unsuccessful deletions of rows from all tables in the database.
- Notify an auditor of actions by an unauthorized user. For example, an unauthorized user could change or delete data, or a user has more privileges than expected, which can lead to reassessing user authorizations.
- Detect problems with an authorization or access control implementation. For example, you can create audit policies that you expect will never generate an audit record because the data is protected in other ways. However, if these policies do generate audit records, then you will know the other security controls are not properly implemented.
- Address auditing requirements for compliance.

- investigate suspicious activity. For example, if an unauthorized user is deleting data from tables, the security administrator might decide to audit all connections to the database and all successful and unsuccessful deletions of rows from all tables in the database.

- monitor and gather data about specific database activities. For example, the database administrator can gather statistics about which tables are being updated, how many logical I/Os are performed, or how many concurrent users connect at peak times.

Types of Auditing

Oracle supports three general types of auditing:

Statement auditing

The selective auditing of SQL statements with respect to only the type of statement, not the specific objects on which it operates. Statement auditing options are typically broad, auditing the use of several types of related actions per option; for example, AUDIT TABLE, which tracks several DDL statements regardless of the table on which they are issued. You can set statement auditing to audit selected users or every user in the database.

Privilege auditing

The selective auditing of the use of powerful system privileges to perform corresponding actions, such as AUDIT CREATE TABLE. Privilege auditing is more focused than statement auditing, auditing only the use of the target privilege. You can set privilege auditing to audit a selected user or every user in the database.

Object auditing

The selective auditing of specific statements on a particular schema object, such as AUDIT SELECT ON EMP. Object auditing is very focused, auditing only a specific statement on a specific object. Object auditing always applies to all users of the database.

You can set audit options to determine the type of audit information that is collected. Oracle allows audit options to be focused or broad in the following areas:

- audit successful statement executions, unsuccessful statement executions, or both
- audit statement executions once per user session or once every time the statement is executed
- audit activities of all users or of a specific user

Audit Records and the Audit Trail

Audit records include such information as the operation that was audited, the user performing the operation, and the date/time of the operation. Audit records can be stored in either a data dictionary table, called the audit trail, or an operating system audit trail.

The database audit trail is a single table named AUD$ in the SYS schema of each Oracle database's data dictionary..

Depending on the events audited and the auditing options set, the audit trail records can contain different types of information. The following information is always included in each audit trail record, provided that the information is meaningful to the particular audit action:

- the user name
- the session identifier
- the terminal identifier
- the name of the object accessed
- the operation performed or attempted
- the completion code of the operation
- the date and time stamp
- the system privileges used (including MAC privileges for Trusted Oracle)
- the label of the user session (for Trusted Oracle only)
- the label of the object accessed (for Trusted Oracle only)

Statement Auditing

Statement auditing is the selective auditing of related groups of statements that fall into two categories:

- DDL statements, regarding a particular *type* of database structure or object, but not a specifically named structure or object (for example, AUDIT TABLE audits all CREATE and DROP TABLE statements)
- DML statements, regarding a particular *type* of database structure or object, but not a specifically named structure or object (for example, AUDIT SELECT TABLE audits all SELECT . . . FROM TABLE/VIEW/SNAPSHOT statements, regardless of the table, view, or snapshot)

Statement auditing can be broad and audit the activities of all database users, or focused and audit only the activities of a select list of database users.

Enabling Statement Auditing

Two special cases of statement auditing are discussed in the following sections.

Auditing Connections and Disconnections

The `SESSION` statement option is unique because it does not generate an audit record when a particular type of statement is issued. This option generates a single audit record for each session created by connections to an instance. An audit record is inserted into the audit trail at connect time and updated at disconnect time. Cumulative information about a session is stored in a single audit record that corresponds to the session. This record can include connection time, disconnection time, and logical and physical I/O processed, among other information.

To audit all successful and unsuccessful connections to and disconnections from the database, regardless of user, `BY SESSION` (the default and only value for this option), enter the following statement:

```
AUDIT SESSION;
```

You can set this option selectively for individual users also, as in the next example:

```
AUDIT SESSION

BY jeff, lori;
```

Privilege Auditing

Privilege auditing is the selective auditing of the statements allowed using a system privilege. For example, auditing of the SELECT ANY TABLE system privilege audits users' statements that are executed using the SELECT ANY TABLE system privilege.

You can audit the use of any system privilege. In all cases of privilege auditing, owner privileges and object privileges are checked before the use of system privileges. If these other privileges suffice to permit the action, the action is not audited. If similar statement and privilege audit options are both set, only a single audit record is generated. For example, if the statement option TABLE and the system privilege CREATE TABLE are both audited, only a single audit record is generated each time a table is created.

Privilege auditing is more focused than statement auditing because each option audits only specific types of statements, not a related list of statements. For example, the statement auditing option TABLE audits CREATE TABLE, ALTER TABLE, and DROP TABLE statements, while the privilege auditing option CREATE TABLE audits only CREATE TABLE statements, since only the CREATE TABLE statement requires the CREATE TABLE privilege.

Privilege auditing can be broad, and audit the activities of all database users, or focused, and audit only the activities of a select list of database users.

Enabling Privilege Auditing

Privilege audit options exactly match the corresponding system privileges. For example, the option to audit use of the `DELETE ANY TABLE` privilege is `DELETE ANY TABLE`. To turn this option on, you use a statement similar to the following example:

```
AUDIT DELETE ANY TABLE

    BY ACCESS

    WHENEVER NOT SUCCESSFUL;
```

To audit all successful and unsuccessful uses of the `DELETE ANY TABLE` system privilege, enter the following statement:

```
AUDIT DELETE ANY TABLE;
```

To audit all unsuccessful `SELECT`, `INSERT`, and `DELETE` statements on all tables and unsuccessful uses of the `EXECUTE PROCEDURE` system privilege, by all database users, and by individual audited statement, issue the following statement:

```
AUDIT SELECT TABLE, INSERT TABLE, DELETE TABLE, EXECUTE PROCEDURE

      BY ACCESS

      WHENEVER NOT SUCCESSFUL;
```

The `AUDIT SYSTEM` system privilege is required to set any statement or privilege audit option. Normally, the security administrator is the only user granted this system privilege.

Object Auditing

Object auditing is the selective auditing of specific DML statements (including queries), and GRANT and REVOKE statements for specific schema objects. Object auditing audits the operations permitted by object privileges, such as SELECT or DELETE statements on a given table, as well as the GRANT and REVOKE statements that control those privileges.

You can audit statements that reference tables, views, sequences, *standalone* stored procedures and functions, and packages (procedures in packages cannot be audited individually). Notice that statements that reference clusters, database links, indexes, or synonyms are not audited directly.

You can, however, audit access to these objects indirectly by auditing the operations that affect the base table. Object audit options are always set for all users of the database; these options cannot be set for a specific list of users. Oracle provides a mechanism for setting default object audit options for all auditable schema objects.

Enabling Object Auditing

A user can set any object audit option for the objects contained in the schema of the user. The `AUDIT ANY` system privilege is required to set an object audit option for an object contained in another user schema or to set the default object auditing option. Normally, the security administrator is the only user granted the `AUDIT ANY` privilege.

To audit all successful and unsuccessful `DELETE` statements on the `jeff.emp` table, `BY SESSION` (the default value), enter the following statement:

```
AUDIT DELETE ON jeff.emp;
```

To audit all successful `SELECT`, `INSERT`, and `DELETE` statements on the `dept` table owned by user `jward`, `BY ACCESS`, enter the following statement:

```
AUDIT SELECT, INSERT, DELETE

     ON jward.dept

     BY ACCESS

     WHENEVER SUCCESSFUL;
```

To set the default object auditing options to audit all unsuccessful `SELECT` statements, `BY SESSION` (the default), enter the following statement:

Viewing Database Audit Trail Information

The database audit trail (`SYS.AUD$`) is a single table in each Oracle database data dictionary. Several predefined views are available to present auditing information from this table in a meaningful way. If you decide not to use auditing, then you can later delete these views. The following subsections show you what is in these views, how to use them, and how to delete them:

- Audit Trail Views
- Using Audit Trail Views to Investigate Suspicious Activities
- Deleting the Audit Trail Views

Audit Trail Views

The following views are created upon installation:

View	Description
`STMT_AUDIT_OPTION_MAP`	Contains information about auditing option type codes. Created by the `SQL.BSQ` script `DATABASE` time.
`AUDIT_ACTIONS`	Contains descriptions for audit trail action type codes.
`ALL_DEF_AUDIT_OPTS`	Contains default object-auditing options that will be applied when objects are created.
`DBA_STMT_AUDIT_OPTS`	Describes current system auditing options across the system and by user.
`DBA_PRIV_AUDIT_OPTS`	Describes current system privileges being audited across the system and by user.
`DBA_OBJ_AUDIT_OPTS` `USER_OBJ_AUDIT_OPTS`	Describes auditing options on all objects. The `USER` view describes auditing options on the current user.
`DBA_AUDIT_TRAIL` `USER_AUDIT_TRAIL`	Lists all audit trail entries. The `USER` view shows audit trail entries relating to current us
`DBA_AUDIT_OBJECT`	Contains audit trail records for all objects in the system. The `USER` view lists audit trail concerning objects that are accessible to the current user.

View	Description
USER_AUDIT_OBJECT	
DBA_AUDIT_SESSION USER_AUDIT_SESSION	Lists all audit trail records concerning CONNECT and DISCONNECT. The USER view lists concerning connections and disconnections for the current user.
DBA_AUDIT_STATEMENT USER_AUDIT_STATEMENT	Lists audit trail records concerning GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYS throughout the database, or for the USER view, issued by the user.
DBA_AUDIT_EXISTS	Lists audit trail entries produced BY AUDIT NOT EXISTS.
DBA_AUDIT_POLICIES	Shows all the auditing policies on the system.
DBA_FGA_AUDIT_TRAIL	Lists audit trail records for value-based auditing.
DBA_COMMON_AUDIT_TRAIL	Combines standard and fine-grained audit log records, and includes SYS and mandato in XML format.