1

Overview of Oracle SQL Firewall

SQL Firewall is part of the Oracle Database kernel. Learn about Oracle SQL Firewall and its use cases and features from this section.

1.1 About Oracle SQL Firewall

Oracle SQL Firewall provides real-time protection against common database attacks by restricting database access to only authorized SQL statements or connections for a designated user.

It mitigates risks from SQL injection attacks, anomalous access, and credential theft or abuse, preventing or detecting potential SQL injection attacks.

You can use SQL Firewall to control which SQL statements are allowed to be processed by the database. In addition, SQL Firewall can use session context data such as IP address to restrict database connections. Unauthorized SQL and database connection can be logged and blocked.

SQL Firewall helps to address the following three **use cases**:

- Provide real-time protection by restricting database access to only authorized SQL statements and database connections.
- Mitigate SQL injection attacks, anomalous access, and credential theft/abuse risks.
- Enforce trusted database connection paths.

SQL Firewall offers the following **benefits**:

- SQL Firewall inspects all incoming database connections and SQL statements, including those from PL/SQL, whether local or over the network, encrypted or clear text. It cannot be bypassed. It only allows explicitly authorized SQL. For all other SQL, it logs the offending statements and raises violations. This statement could have been a SQL injection attack or a new SQL statement that the authorized user has not run before.
- You can decide whether you want to block unauthorized SQL or only log it. This gives you the flexibility on how to handle attacks.
- SQL Firewall evaluates the complete SQL and the processing context. By running inside the Oracle database server, the firewall easily handles encoding of the SQL statement, synonyms, dynamically generated object names, and any SQL statements that are dynamically generated in PL/SQL units.
- SQL Firewall relies on the allow-listing (an allow-list is a set of permitted actions) of the authorized SQL statements and associated trusted database connection paths while blocking the rest. You train the SQL Firewall by simply capturing authorized SQL statements for a database account. Subsequently, the firewall detects and prevents unauthorized SQL and potential SQL injection attacks. A typical use case with allow-listed SQL statements is for application SQL workloads issued by application service account.
- SQL Firewall can also block connections that do not come from trusted IP addresses, operating system user names, or program names. This function is useful when you want to put some protection in place immediately, while you create the allow-list of SQL statements for your applications. This feature ensures that any direct access to your databases is

coming exclusively from trusted endpoints. This also helps mitigate the risk of stolen or misused application service account credentials.

SQL Firewall enables you to build an allow-list policy for each database user of SQL statements that a typical database user performs, and then detects, blocks, and logs any unexpected SQL.

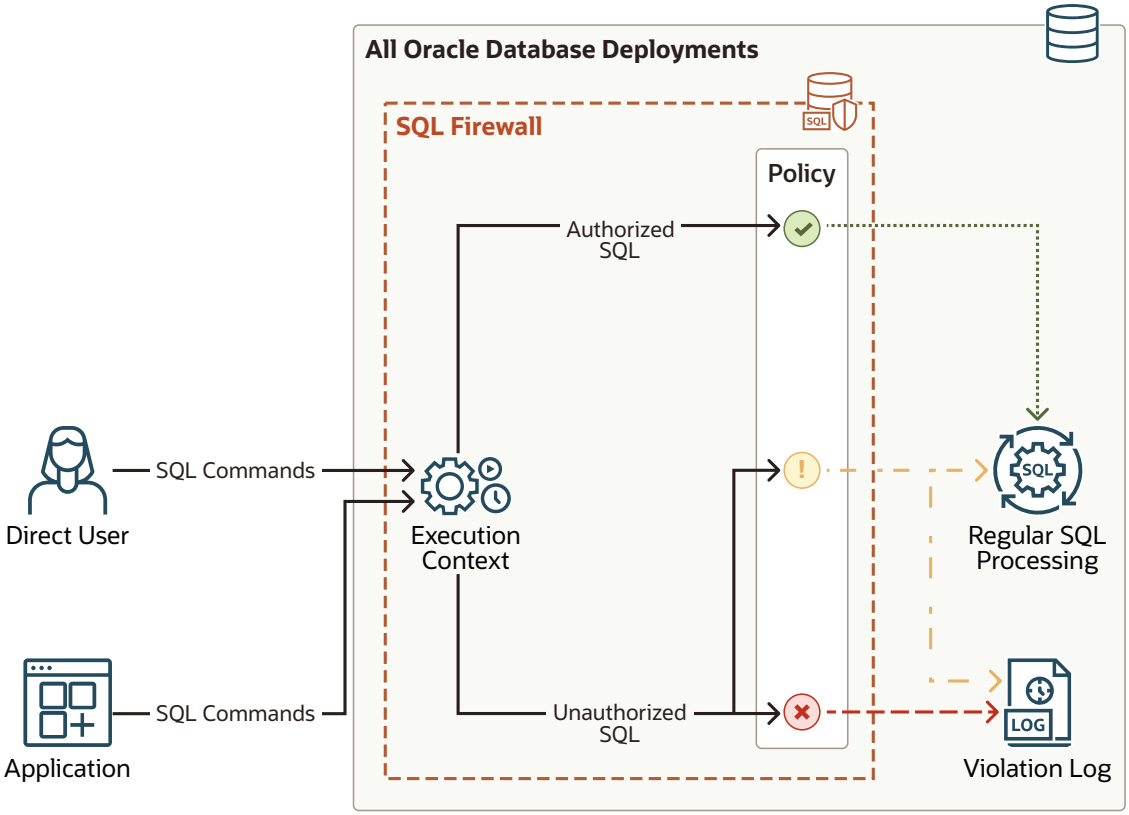
SQL Firewall policies work at a database account level, whether of an application service account or a direct database user, such as a reporting user or a database administrator. In other words, you might have one SQL Firewall policy for the database user `HR` and another for the database user `pfitch`. This flexibility allows you to gradually build up the protection level of the database, starting from either the database administrators or the application service accounts.

You can use SQL Firewall in both the root and a pluggable database (PDB). SQL Firewall is a simple and easy-to-use firewall solution for all Oracle Database deployments, such as on-premises, cloud, multitenant, Oracle Data Guard, or Oracle Real Application Clusters. SQL Firewall works in conjunction with other Oracle Database security features such as Transparent Data Encryption (TDE), database auditing, and Oracle Database Vault.

SQL Firewall supports (that is, it captures and enforces on) all SQL commands except transaction control commands (`SAVEPOINT`, `COMMIT`, and `ROLLBACK`). Additionally, SQL Firewall supports the SQL*Plus commands `PASSWORD` and `DESCRIBE`, and remote procedure calls (RPC) through database links.

The following diagram explains how SQL Firewall operates inline within the Oracle Database kernel.

Figure 1-1 SQL Firewall Process



1. A user logs in to the Oracle database through a web application.
2. The user runs SQL statements, creating inbound traffic to the Oracle database.
3. SQL Firewall inspects the incoming database connections and SQL statements, and enforces it against the permitted SQL statements and trusted connection paths in the allow-list policy for the user. SQL Firewall's processing outcome is one of the following options:
   - Allow the SQL for its subsequent execution.
   - Allow the SQL and log it.
   - Log and optionally block unauthorized SQL.

1.2 Licensing Oracle SQL Firewall

Oracle SQL Firewall must be licensed for use. There are two paths to its license.

- **Included with Oracle Database Vault.** Oracle Database Vault is an extra-cost option of Oracle Database. See *Oracle Database Licensing Information User Manual*.
- **Included with Oracle Audit Vault and Database Firewall (AVDF).** AVDF is a separate Oracle product and requires a license. See *Oracle Database Licensing Information User Manual*.

1.3 Getting Started with Oracle SQL Firewall

To get started with Oracle SQL Firewall, you follow three steps: first, enable Oracle SQL Firewall; second, capture the user's normal SQL activities; and third, enable and enforce allow-lists.

1. **Enable SQL Firewall.** As an administrator with appropriate privileges, enable SQL Firewall in the Oracle database.
2. **Capture the normal SQL activities.** For every database user that you want to protect with SQL Firewall, you must enable SQL Firewall to learn the normal SQL traffic of the database user. It does this by capturing all the authorized SQL statements over trusted database connection paths. You can query SQL Firewall-specific data dictionary views to review this captured data to determine if the collected SQL statements and connection paths is adequate to constitute the allow-lists.
   After you review the captured SQL statements, you can generate a SQL Firewall policy with allow-lists that set the baseline for allowed SQL statements and allowed contexts. Allowed SQL statements constitute the approved SQL statements. At run-time, when the policy is enforced, any incoming SQL queries that have a structure syntactically similar to the SQL signature in the policy allow-list will be passed for execution if the corresponding run-time execution context also meets the set of allowed contexts. Allowed contexts represent trusted database connection paths and consist of three distinct groups—client IP addresses, operating system program names, and operating system user names. When the user connects to the database, SQL Firewall checks the current session context attributes, and ensures that access to the database comes exclusively from trusted endpoints defined in the allow-lists. You can review the allow-list and make modifications by using the `DBMS_SQL_FIREWALL` procedures any time.
3. **Enable and enforce the allow-lists.** Enabling the generated SQL Firewall policy protects the database user. SQL Firewall enforces and checks the allow-lists when the user connects to the database and issues SQL statements. You can let SQL Firewall know if you want to enforce checks on allowed contexts, allowed SQL statements, or both. If the database connection paths and SQL statements in the incoming SQL traffic do not match

the entries in the enabled and enforced allow-lists, then a SQL Firewall violation is triggered and this incident is logged in the violation log. You can let SQL Firewall know how to respond to SQL Firewall violation incident: allow the traffic to proceed to the database or block. Blocking raises an `ORA-47605: SQL Firewall violation` error, which prevents anomalous database access, without disrupting client connections for SQL violations following a mismatch of SQL statements. However, blocking for context violations will disrupt and terminate client connections following a mismatch of contexts.
SQL Firewall raises and logs violations in real-time for every unmatched scenario of database connection or SQL command execution against the entries in the enabled allow-lists of the SQL Firewall policy. A security administrator can monitor the SQL Firewall violation log `DBA_SQL_FIREWALL_VIOLATIONS` to detect the presence of these abnormalities. You may want to audit SQL Firewall violations (especially the blocked ones); their occurrence potentially indicates abnormal database access attempts including SQL Injection and credential theft or abuse. Auditing violations places a record of the violation in the database audit trail, where it can be protected from tampering.

Key points to consider are as follows:

- Oracle Database mandatorily audits all SQL Firewall administrative actions and writes these to the unified audit trail data dictionary view, `UNIFIED_AUDIT_TRAIL`. You can also create unified audit policies to monitor SQL Firewall violations. Another way to monitor and troubleshoot SQL Firewall is to use the `SQL_FIREWALL` trace file setting.
- You can export and import SQL Firewall metadata, including existing allow-lists, by using the Oracle Data Pump `EXPDB` and `IMPDB` utilities.
- Oracle recommends that you periodically monitor and purge violations logs by using the `DBMS_SQL_FIREWALL.PURGE_LOG` procedure as part of routine SQL Firewall management tasks. In a well trained environment, violation logs are not expected to be voluminous.
- SQL Firewall captures SQL statements that the user issues directly or from PL/SQL units that the user invokes in sessions of target users.
- SQL Firewall captures only SQL statements that are executed successfully. That is, if a SQL statement fails to execute due to any error, SQL Firewall does not capture the corresponding statement.
- SQL Firewall captures SQL statements before any internal query transformation (for example, views or macro expansions, or Oracle Virtual Private Database policy enforcement) is performed.
- SQL Firewall normalizes captured SQL statements and replaces literal values with special symbols before storing them in the log tables.
- The session context attributes (client IP address, operating system user name, and operating system program name) are checked only once during session creation.
- You can append to the existing allow-list anytime by using either the `DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST` procedure or the `DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST_SINGLE_SQL` procedure from the following two sources:
  - Violation log: `DBA_SQL_FIREWALL_VIOLATIONS` data dictionary view
  - Capture log: `DBA_SQL_FIREWALL_CAPTURE_LOGS` data dictionary view
- For existing sessions that were created before the allow-list is enabled, SQL Firewall also checks the allowed contexts, but does not terminate existing sessions even if they have unmatched session contexts. In this case, SQL Firewall does not log the violation.

1.4 Privileges for Configuring and Using Oracle SQL Firewall

You must be granted the appropriate role to administer Oracle SQL Firewall or to query the views that are associated with Oracle SQL Firewall.

To administer Oracle SQL Firewall, you must be granted the `SQL_FIREWALL_ADMIN` role. This role provides the following privileges:

- The `ADMINISTER SQL FIREWALL` system privilege, which is required to run the PL/SQL procedures in the `DBMS_SQL_FIREWALL` package
- The `EXECUTE` privilege for the `DBMS_SQL_FIREWALL` PL/SQL package
- The `READ` privilege for the SQL Firewall `DBA_SQL_FIREWALL_*` data dictionary views

To be able to query the `DBA_SQL_FIREWALL_*` data dictionary views (but not administer SQL Firewall), users must be granted the `SQL_FIREWALL_VIEWER` role.

> **Note:**
>
> The SQL Firewall `SQL_FIREWALL_ADMIN` and `SQL_FIREWALL_VIEWER` roles are powerful roles. Only grant these roles to trusted users.

Related Topics

- Oracle SQL Firewall Data Dictionary Views
  Oracle Database provides a set of data dictionary views that provide information about Oracle SQL Firewall configurations.

1.5 Getting Hands-On Experience with Oracle SQL Firewall

You can use the Oracle LiveLabs workshop for Oracle SQL Firewall to get experience using SQL Firewall.

See the following LiveLabs:

- Get Started with Oracle Data Safe Fundamentals which includes a section on SQL Firewall
- DB Security - SQL Firewall

The following sample demonstration scripts and video of Oracle SQL Firewall in action are also provided for your reference

- Oracle SQL Firewall sample demo scripts