2
Configuring Oracle SQL Firewall

You can configure Oracle SQL Firewall in either an Oracle database using the DBMS_SQL_FIREWALL package, or you can configure it in Oracle Data Safe.

- **DBMS_SQL_FIREWALL Package:** Use the PL/SQL procedures in the DBMS_SQL_FIREWALL package to manage SQL Firewall within an individual Oracle Database instance.
- **Oracle Data Safe:** You can use the Data Safe user interface if you want to manage multiple SQL Firewalls centrally. You can use Data Safe REST APIs, software developer kits (SDKs), CLI, and Terraform for further automation and integration. You can also use the more extensive Oracle Cloud Infrastructure (OCI) ecosystem for integrating SQL Firewall violations with its alerts and notifications.

2.1 Configuring and Managing Oracle SQL Firewall with the DBMS_SQL_FIREWALL Package

After you configure Oracle SQL Firewall for a target user, you can perform maintenance tasks such as modifying the configuration, purging old logs, and troubleshooting errors.

2.1.1 Configuring Oracle SQL Firewall Using the DBMS_SQL_FIREWALL Package

A user who has the SQL_FIREWALL_ADMIN role can use the DBMS_SQL_FIREWALL PL/SQL package to configure Oracle SQL Firewall in the root or a pluggable database (PDB).

1. Connect to the root or PDB as a user who has been granted the SQL_FIREWALL_ADMIN role.
2. Enable SQL Firewall.

   ```
   EXEC DBMS_SQL_FIREWALL.ENABLE;
   ```

3. For every database user to protect with SQL Firewall in the Oracle database, enable SQL Firewall to learn the normal SQL traffic of the database user by capturing all the authorized SQL statements over trusted database connection paths.

   The examples in this procedure assume the user is a PDB user named APP. For example:

   ```
   BEGIN
     DBMS_SQL_FIREWALL.CREATE_CAPTURE (
       username         => 'APP',
       top_level_only   => TRUE,
       start_capture    => TRUE
     );
   END;
   /
   ```

   In this specification:

- username is the name of the application user that SQL Firewall will monitor. You can only create one capture for each user. You cannot create SQL Firewall captures for the SYS, SYSDG, SYSBACKUP, SYSRAC, SYSKM, DVSYS, LBACSYS, or AUDSYS users.
- top_level_only controls the level of SQL statements that are captured.
  - TRUE generates capture logs only for top-level SQL statements, that is, statements that the user directly runs.
  - FALSE generates capture logs for both top-level SQL statements and SQL commands issued from PL/SQL units. The default is FALSE.
- start_capture controls when the capture will be effective.
  - TRUE enables SQL Firewall to start capturing the target user's activities right away. The default is TRUE.
  - FALSE creates a capture for the user, but does not start the capture right away. When you want to start the capture later on, you must run the DBMS_SQL_FIREWALL.START_CAPTURE procedure for the user. For example:

    ```
    EXEC DBMS_SQL_FIREWALL.START_CAPTURE ('APP');
    ```

As an application service account, run the normal application SQL workload from the trusted database connection paths when the capture is started for the application service account. In the event of a change in application in the SQL workload following application patching, you may want SQL Firewall to unlearn and learn, starting over. You can delete the current capture, and create a new one. Specifically, if you want to restart the capture process, then you must first stop this capture (if it is started), then either purge the capture logs and start this capture again, or, delete this capture and create (and start) the capture again.

4. Review the capture logs and sessions logs to determine the adequacy of the capture.

   For example:

   ```
   SELECT SQL_TEXT FROM DBA_SQL_FIREWALL_CAPTURE_LOGS WHERE USERNAME = 'APP';
   ```

5. Stop the capture.

   For example:

   ```
   EXEC DBMS_SQL_FIREWALL.STOP_CAPTURE ('APP');
   ```

6. Generate the SQL Firewall policy with allow-lists for the user:

   A SQL Firewall policy with allow-lists sets the baseline for allowed SQL statements and allowed contexts. Allowed SQL statements constitute the approved SQL statements. Allowed contexts represent trusted database connection paths. SQL Firewall creates the allow-list based on data collected from existing capture logs for the user.
   For example:

   ```
   EXEC DBMS_SQL_FIREWALL.GENERATE_ALLOW_LIST ('APP');
   ```

7. To find the permitted and allowed SQL statements that the user can run, query the DBA_SQL_FIREWALL_ALLOWED_* data dictionary views.

For example:

```
SELECT SQL_TEXT FROM DBA_SQL_FIREWALL_ALLOWED_SQL WHERE USERNAME = 'APP';
```

To find the trusted database connection paths for the user, perform the following queries:

```
SELECT OS_PROGRAM FROM DBA_SQL_FIREWALL_ALLOWED_OS_PROG WHERE USERNAME = 
'APP';
```

```
SELECT OS_USER FROM DBA_SQL_FIREWALL_ALLOWED_OS_USER WHERE USERNAME = 
'APP';
```

```
SELECT IP_ADDRESS FROM DBA_SQL_FIREWALL_ALLOWED_ALLOWED_IP_ADDR WHERE 
USERNAME = 'APP';
```

8. Optionally, add or modify entries in the allowed contexts by running the `DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT` and `DBMS_SQL_FIREWALL.DELETE_ALLOWED_CONTEXT` procedures.

   You can only add a context after you have generated the allow-list. A context can specify the client IP address, names of operating system users, or the operating system program that can be used for database connections. You can add as many context values as you need. For example, if the user's allowed context list does not contain the IP address 192.0.2.1 but you want to allow the user to connect from this IP after the enablement of the allow-list:

```
BEGIN
  DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT (
    username       => 'APP',
    context_type   => DBMS_SQL_FIREWALL.IP_ADDRESS,
    value          => '192.0.2.1'
   );
END;
/
```

   To specify all possibilities for a specific `context_type`, enter the `%` wildcard.

   The following three types of `context_type` settings are valid:

   - `DBMS_SQL_FIREWALL.IP_ADDRESS` accepts IPv4 and IPv6 addresses and subnets in the CIDR notation. It accepts the value `Local` (case sensitive) for local connections when the IP address is not available.
   - `DBMS_SQL_FIREWALL.OS_USERNAME` accepts any valid operating system user name, such as `oracle`.
   - `DBMS_SQL_FIREWALL.OS_PROGRAM` accepts any valid operating system program name that the user uses to run SQL statements, such as `sqlplus` or `SQL Developer`.

   You can query the following data dictionary views to check the contexts:

   - `DBA_SQL_FIREWALL_ALLOWED_IP_ADDR`
   - `DBA_SQL_FIREWALL_ALLOWED_OS_USER`
   - `DBA_SQL_FIREWALL_ALLOWED_OS_PROG`

9. Enable the generated SQL Firewall policy to protect the database user.

   The SQL Firewall enforces checks on the allow-lists when the user connects to the database and issues SQL statements.

   This enablement becomes effective immediately, even in the existing sessions of the target user.
   For example:

   ```
   BEGIN
     DBMS_SQL_FIREWALL.ENABLE_ALLOW_LIST (
       username       => 'APP',
       enforce        => DBMS_SQL_FIREWALL.ENFORCE_SQL,
       block          => TRUE
      );
   END;
   /
   ```

   In this specification:

   - `username` can be a specific user whose allow-list has been generated, or it can be all users whose allow-list are not currently enabled. To specify all users, use `NULL` as the value.
   - `enforce` specifies one of the following enforcement types:
     - `DBMS_SQL_FIREWALL.ENFORCE_CONTEXT` enforces the allowed contexts that have been configured.
     - `DBMS_SQL_FIREWALL.ENFORCE_SQL` enforces the allowed SQL that has been configured.
     - `DBMS_SQL_FIREWALL.ENFORCE_ALL` enforces both allowed contexts and allowed SQL. This setting is the default.
   - `block` specifies the following:
     - `TRUE` blocks the user's database connection or the user's SQL execution whenever the user violates the allow-list definition.
     - `FALSE` allows unmatched user database connections or SQL commands to proceed. This setting is the default.

     SQL Firewall always generates a violation log for any unmatched user database connection or SQL statement regardless of the enforcement option.

     At this stage, if the user attempts to perform a SQL query that violates the allow-list and you have specified SQL Firewall to block this SQL, then an `ORA-47605: SQL Firewall violation` error appears.

10. Monitor the violation log for abnormal SQL connection attempts or SQL queries that are reported if they are not in allow-list.

    For example:

    ```
    SELECT SQL_TEXT, FIREWALL_ACTION, IP_ADDRESS, CAUSE, OCCURRED_AT
    FROM DBA_SQL_FIREWALL_VIOLATIONS WHERE USERNAME = 'APP';
    ```

Output similar to the following appears:

```
SQL_TEXT                                                  FIREWALL_ACTION
IP_ADDRESS   CAUSE            OCCURRED_AT
--------------------------------------------------------- ----------------
-----------  ----------------- -----------------------------------

SELECT SALARY FROM HR.EMPLOYEES WHERE SALARY >:"SYS_B_0"  BLOCKED
192.0.2.146  Context violation 12-MAY-23 11.12.39.626053 PM +00:00
```

Related Topics

- Configuring and Managing Oracle SQL Firewall with the DBMS_SQL_FIREWALL Package
  After you configure Oracle SQL Firewall for a target user, you can perform maintenance tasks such as modifying the configuration, purging old logs, and troubleshooting errors.
- Oracle SQL Firewall Data Dictionary Views
  Oracle Database provides a set of data dictionary views that provide information about Oracle SQL Firewall configurations.

2.1.2 Modifications to Oracle SQL Firewall Configurations

After you create an Oracle SQL Firewall configuration for a user, you can modify the configuration as necessary.

To find information about Oracle SQL Firewall configurations, you can query the DBA_SQL_FIREWALL_* data dictionary views.

Table 2-1 lists operations that you can perform after you have configured SQL Firewall.

Table 2-1 Oracle SQL Firewall Modification Procedures

Operation	Procedure
Enable SQL Firewall	• To enable SQL Firewall in the database, use DBMS_SQL_FIREWALL.ENABLE.
Manage captures	• To create a capture, use DBMS_SQL_FIREWALL.CREATE_CAPTURE. • To start a capture, use DBMS_SQL_FIREWALL.START_CAPTURE. • To modify a capture, delete the current one by using DBMS_SQL_FIREWALL.DROP_CAPTURE, and then create a new one by using DBMS_SQL_FIREWALL.CREATE_CAPTURE. • To stop the SQL Firewall capture for the specified user, use DBMS_SQL_FIREWALL.STOP_CAPTURE. • To delete the SQL Firewall capture for a specified user and delete all the existing capture logs for this user: 1. Use DBMS_SQL_FIREWALL.STOP_CAPTURE to stop the capture process. 2. Use DBMS_SQL_FIREWALL.DROP_CAPTURE to remove the capture.

Table 2-1 (Cont.) Oracle SQL Firewall Modification Procedures

Operation	Procedure
Manage allow-lists	• To generate an allow-list for a given user, use `DBMS_SQL_FIREWALL.GENERATE_ALLOW_LIST`. • To enable an allow-list for a given user, use `DBMS_SQL_FIREWALL.ENABLE_ALLOW_LIST`. • To update an allow-list enforcement, use `DBMS_SQL_FIREWALL.UPDATE_ALLOW_LIST_ENFORCEMENT`. • To prevent SQL Firewall from capturing and enforcing allow-lists for database connections and SQL executions in Oracle Scheduler jobs, use `DBMS_SQL_FIREWALL.EXCLUDE`. • To append all the SQL from a capture log or violation log (or from both) to the allow-list, use the `DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST` procedure. You can run this procedure when the allow-list is either enabled or disabled. The change takes place immediately. • To append a single SQL record from a capture log or violation log to the allow-list, use the `DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST_SINGLE_SQL` procedure as follows: 1. Query the `DBA_SQL_FIREWALL_VIOLATIONS` or the `DBA_SQL_FIREWALL_CAPTURE_LOGS` data dictionary view to find the target SQL record that you want to add to the allow-list. 2. Enter the obtained `USERNAME`, `SQL_SIGNATURE`, `CURRENT_USER`, and `TOP_LEVEL` values of that record in the `DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST_SINGLE_SQL` procedure to add the target SQL record to the allow-list. You can run `DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST_SINGLE_SQL` when the allow-list is either enabled or disabled. The change takes place immediately. • To export the allow-list of a given user to JSON format into the specified CLOB, use `DBMS_SQL_FIREWALL.EXPORT_ALLOW_LIST`. • To import the allow-list for a given user into a target database, use `DBMS_SQL_FIREWALL.IMPORT_ALLOW_LIST`. • To disable an allow-list for a given user, use `DBMS_SQL_FIREWALL.DISABLE_ALLOW_LIST`. • To add or delete any context values from allowed context lists, use `DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT` or `DBMS_SQL_FIREWALL.DELETE_ALLOWED_CONTEXT`, respectively. • To delete any SQL statement from allowed SQL lists, use `DBMS_SQL_FIREWALL.DELETE_ALLOWED_SQL`. • To delete the allow-list for a specified user: 1. Disable the allow-list by using `DBMS_SQL_FIREWALL.DISABLE_ALLOW_LIST`. 2. Use `DBMS_SQL_FIREWALL.DROP_ALLOW_LIST`.
Manage allowed contexts	• To add a specified value to the allowed contexts of a specified user for the given context type, use `DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT`. • To modify an allowed context, delete the current one by using `DBMS_SQL_FIREWALL.DELETE_ALLOWED_CONTEXT`, and then create a new one by using `DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT`. • To delete the specified value from the allowed contexts of a specified user for the given context type, use `DBMS_SQL_FIREWALL.DELETE_ALLOWED_CONTEXT`.

Table 2-1 (Cont.) Oracle SQL Firewall Modification Procedures

Operation	Procedure
Manage allowed SQL	• To delete the specified entry from the allowed SQL of a specified user, use `DBMS_SQL_FIREWALL.DELETE_ALLOWED_SQL`. You can run this procedure when the allow-list is either enabled or disabled, and the change takes place immediately.
Manage SQL Firewall log tables	• To move the SQL Firewall log tables to a different user-defined tablespace other than the default tablespace, `SYSAUX`: 1. Disable SQL Firewall by using `DBMS_SQL_FIREWALL.DISABLE`. 2. Use the `MOVE` clause of the `ALTER TABLE` statement to perform the move operation. You can also use the `DBMS_SQL_FIREWALL.MOVE_LOG_TABLE` procedure to move the SQL Firewall log tables to another tablespace. • To purge capture logs or violation logs for a user or all users, use `DBMS_SQL_FIREWALL.PURGE_LOG`. • To flush all the SQL Firewall logs that reside in the memory into the log tables, use `DBMS_SQL_FIREWALL.FLUSH_LOGS`.
Disable SQL Firewall	• To disable SQL Firewall in the database and stop all the existing captures and allow-lists that are enabled, use `DBMS_SQL_FIREWALL.DISABLE`.

Related Topics

- *Oracle Database PL/SQL Packages and Types Reference*
- Oracle SQL Firewall Data Dictionary Views
  Oracle Database provides a set of data dictionary views that provide information about Oracle SQL Firewall configurations.

2.1.3 Managing Performance for Capture Logs

Depending on application workloads, Oracle SQL Firewall may generate a large volume of capture logs.

To minimize the adverse impact on database performance, Oracle SQL Firewall relies internally on Fast Ingest for better write performance if sufficient memory is available. To make full use of SQL Firewall, Oracle recommends that you do the following:

- Allocate at least an additional 2G to the `LARGE_POOL_SIZE` parameter setting, on top of the existing `LARGE_POOL_SIZE` requirement.
- Resize the `SGA_TARGET` parameter setting to include this additional requirement. Ensure that the final size is 8G or more.

Related Topics

- *Oracle Database Performance Tuning Guide*

2.1.4 Purging Oracle SQL Firewall Logs

Periodically, you should purge the logs that Oracle SQL Firewall generates by using the `DBMS_SQL_FIREWALL.PURGE_LOG` procedure.

SQL Firewall generates and stores the violation logs in a log table. In an ideal SQL Firewall trained environment, the violation log is not expected to be large. Oracle recommends that you periodically purge these logs. After you verify that the generated allow-list is valid, you should purge unnecessary logs to reclaim the disk space that the logs are using.

1. Log in to the root or the pluggable database (PDB) where SQL Firewall is configured as a user who has been granted the `SQL_FIREWALL_ADMIN` role.
2. Optionally, as a user who has the `SELECT ANY DICTIONARY` system privilege, query the following data dictionary views to check the logs that you plan to purge:
   - `DBA_SQL_FIREWALL_CAPTURE_LOGS`
   - `DBA_SQL_FIREWALL_VIOLATIONS`
3. Connect to the PDB a user who has been granted the `SQL_FIREWALL_ADMIN` role.
4. Run the `DBMS_SQL_FIREWALL.PURGE_LOG` procedure.

   For example:

   ```
   BEGIN
     DBMS_SQL_FIREWALL.PURGE_LOG (
       username     => 'APP',
       purge_time   => '2023-02-01 00:00:00.00 -08:00',
       log_type     => 'DBMS_SQL_FIREWALL.ALL_LOGS'
      );
    END;
   /
   ```

   In this specification:

   - `username` is the target user for which this SQL Firewall configuration was created. If you omit this value, then Oracle Database purges all logs that match the `purge_time` and `log_type` settings.
   - `purge_time` is the timestamp (in `TIMESTAMP` format) that you can specify to purge only logs that were generated before a certain time. If you omit this value, then Oracle Database purges all logs, regardless of the time when they were generated.
   - `log_type` is the type of the logs to be purged. If you do not specify a value, then the default is `DBMS_SQL_FIREWALL.ALL_LOGS`. Specify one of the following constants:
     - `DBMS_SQL_FIREWALL.CAPTURE_LOG`
     - `DBMS_SQL_FIREWALL.VIOLATION_LOG`
     - `DBMS_SQL_FIREWALL.ALL_LOGS` (default)

Related Topics

- *Oracle Database Reference*

2.1.5 Auditing Oracle SQL Firewall Violations by Using Unified Audit Policies

Oracle recommends that you audit SQL Firewall violations as violations indicate the occurrence of potential abnormal database access patterns.

Auditing SQL Firewall violations with unified auditing records the violation in the database audit trail, `UNIFIED_AUDIT_TRAIL` data dictionary view. It is important that you turn on violation auditing after SQL Firewall is fully trained and the allow-lists of the user is complete, to avoid false positives and reduce unnecessary audit volume.

You can create unified audit policies that are specific to SQL Firewall by specifying the `SQL_FIREWALL` component when you create the unified audit policy. When you query the `UNIFIED_AUDIT_TRAIL`, you can query the `FW_ACTION_NAME` and `FW_RETURN_CODE` columns.

> **Note:**
>
> Oracle Database mandatorily audits all invocations of the SQL Firewall `DBMS_SQL_FIREWALL` PL/SQL administrative procedures.

Related Topics

-

2.1.6 Troubleshooting Oracle SQL Firewall by Enabling or Disabling SQL Firewall Trace Files

As a user who has been granted the `ALTER SESSION` or `ALTER SYSTEM` system privilege, you can generate trace files within the PDB in which you are using Oracle SQL Firewall.

You can set SQL Firewall trace events in both the CDB and in individual PDBs.

- To enable tracing for SQL Firewall, use one of the following statements:

  ```
  ALTER SESSION SET EVENTS 'TRACE[SQL_FIREWALL] DISK=trace_level';
  ALTER SYSTEM SET EVENTS 'TRACE[SQL_FIREWALL] DISK=trace_level';
  ```

  In this specification, replace *`trace_level`* with one of the following values:

  - `LOW` shows the minimum tracing information.
  - `HIGH` shows more detailed tracing information, plus the information returned by `LOW`.
  - `HIGHEST` shows the most detailed tracing information, plus the information returned by `HIGH` and `LOW`.

- To disable tracking for SQL Firewall, use one of the following statements:

  ```
  ALTER SESSION SET EVENTS 'TRACE[SQL_FIREWALL] OFF';
  ALTER SYSTEM SET EVENTS 'TRACE[SQL_FIREWALL] OFF';
  ```

Related Topics

-

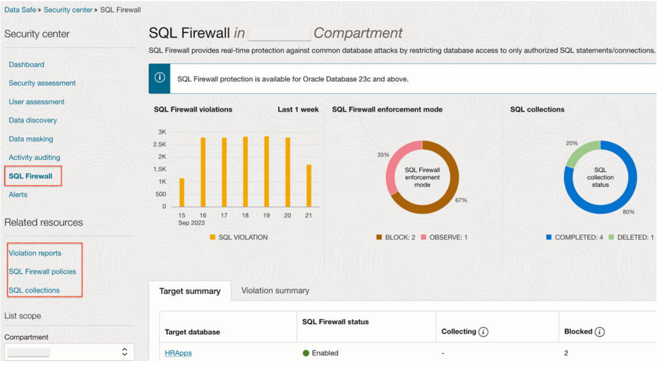
2.2 Configuring and Managing Oracle SQL Firewall with Oracle Data Safe

With Oracle Data Safe on Oracle Cloud, you can manage multiple SQL Firewalls centrally and get a comprehensive view of SQL Firewall violations across a fleet of Oracle databases.

SQL Firewall administrators can use Data Safe to collect SQL activities of a database user with its associated database connection paths (IP address, OS program, OS user), and monitor the progress of the collection. Data Safe enables you generate and enable the SQL Firewall policy from the collected SQL traffic. Data Safe automatically collects the violation logs, and lets you monitor SQL Firewall violations from the console.

The following image shows the SQL Firewall dashboard in Data Safe.

Figure 2-1   SQL Firewall Dashboard in Data Safe



SQL Firewall *in* ______ *Compartment*

SQL Firewall provides real-time protection against common database attacks by restricting database access to only authorized SQL statements/connections.

SQL Firewall protection is available for Oracle Database 23c and above.

Target database	SQL Firewall status	Collecting	Blocked
HRApps	● Enabled	-	2

The violation summary in the dashboard provides a comprehensive view of SQL Firewall violations from all the targets in the compartment that have SQL Firewall enabled for the chosen period. From here, you can drill down into the violations for detailed analysis.

Related Topics

- Start Using SQL Firewall
  In order to begin using SQL Firewall you need to complete the following steps. Ensure you have already completed the prerequisites before starting these steps.

2.2.1 SQL Firewall Overview

The SQL Firewall feature in Oracle Data Safe lets you administer and monitor SQL Firewall across your fleet of Oracle Database 23ai targets.

2.2.1.1 About SQL Firewall

The SQL Firewall feature in Oracle Data Safe lets you administer and monitor SQL Firewall across your fleet of Oracle Database targets. SQL Firewall is a new security feature built into the Oracle Database 23ai kernel and offers protection against risks such as SQL injection attacks and compromised accounts. SQL Firewall inspects all incoming database connections and SQL statements, including the ones from PL/SQL units, whether local or over the network, encrypted or clear text. It only allows explicitly authorized SQL and can log or block SQL statements and connections that do not fall within the SQL Firewall allowlists.

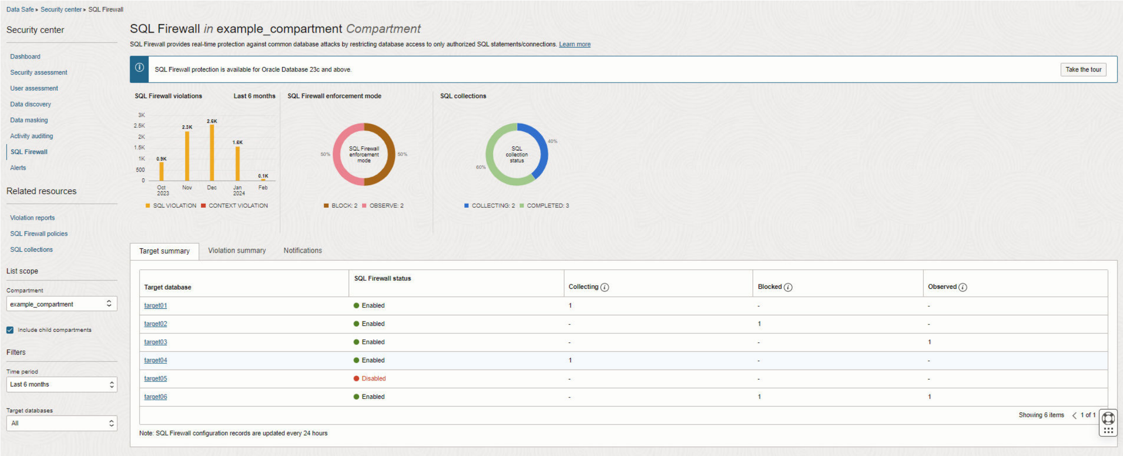
SQL Firewall uses allowlists of authorized SQL statements and trusted database connection paths to determine which SQL statements and connection paths are authorized and which ones should be either logged or blocked. SQL Firewall allowlist policies work at a database account level. You create an SQL Firewall allowlist for a database account by capturing or collecting the expected application SQL workload from expected database connections. Subsequently, the firewall detects and prevents unauthorized SQL and potential SQL injection attacks.

To learn more about SQL Firewall in Oracle Database see the *Oracle Database Oracle SQL Firewall User's Guide*

SQL Firewall can be managed in multiple ways. The PL/SQL procedures in `SYS.DBMS_SQL_FIREWALL` package lets you manage SQL Firewall directly in an Oracle Database (23ai or above). Consider Oracle Data Safe if you are looking forward to leveraging the convenience of the Oracle Cloud Infrastructure (OCI) ecosystem and want to manage and monitor SQL Firewall for a fleet of Oracle Database targets.

Administrators can use Data Safe to collect SQL activities of database accounts, monitor the collection progress, create SQL Firewall policies with allowlist rules (allowed contexts and allowed SQL statements) from the collected SQL activities, and enable SQL Firewall policies. Once a SQL Firewall policy is enabled, Data Safe automatically collects the firewall violation logs from the database and stores them in Data Safe. Those logs are then available for online analysis and reporting across your database fleet as shown in Figure 2-2. You can leverage the Data Safe REST APIs, SDKs, CLI, and Terraform for further automation and integration. You can also leverage the larger OCI ecosystem for integrating SQL Firewall violations with its alerts and notifications.

Figure 2-2   SQL Firewall dashboard in Data Safe



SQL Firewall *in* example_compartment *Compartment*

SQL Firewall provides real-time protection against common database attacks by restricting database access to only authorized SQL statements/connections. Learn more

ⓘ SQL Firewall protection is available for Oracle Database 23c and above.

Target database	SQL Firewall status	Collecting ⓘ	Blocked ⓘ	Observed ⓘ
target01	● Enabled	1	-	-
target02	● Enabled	-	1	-
target03	● Enabled	-	-	1
target04	● Enabled	1	-	-
target05	● Disabled	-	-	-
target06	● Enabled	-	1	1

Note: SQL Firewall configuration records are updated every 24 hours

2.2.1.2 Terms in SQL Firewall

The following terms are used throughout Oracle Data Safe's SQL Firewall feature.

- Database security configuration - This resource represents the target database configurations. Included in the Database security configurations are the SQL Firewall configurations such as the status of the firewall, the time that the firewall status was last updated, violation log auto purge settings, and so on.
- Session context - This represents client information initiating SQL traffic: client IP address, OS program name, and OS username.
- SQL collection - This resource represents the SQL collection for a specific database user in a target database. SQL collection encapsulates the SQL commands issued in the user’s database sessions and their session context.
- SQL Firewall policy - An allowlist policy specific to a database user through which incoming SQL statements will be evaluated to determine if they can take action on the target database. SQL statements can be allowed or, if they're not part of the allowlist, allowed and logged or blocked and logged. The policy can consist only of session context information, only of specific SQL statements, or both.
- SQL violations - This represents SQL statements that were initiated on the target database that are not included in the allowlist in the SQL Firewall policy.

- Context violations - This represents session context from which SQL statements were initiated on the target database that are not included in the allowlist in the SQL Firewall policy.
- Observe and log violations - A SQL Firewall policy enforcement option where initiated SQL statements that are either SQL or context violations are allowed to execute on the target database and the statements and context are logged for later reference.
- Block and log violations - A SQL Firewall policy enforcement option where initiated SQL statements that are either SQL or context violations are blocked and can't execute on the target database. The statements and context are logged for later reference.

2.2.1.3 Prerequisites for SQL Firewall

SQL Firewall requires you to register an Oracle Database 23ai target database in Data Safe. Users must be granted specific permissions in IAM.

These are the prerequisites for using the SQL Firewall feature in Data Safe:

- Register an Oracle Database 23ai or later. For more information see, Target Registration in the *Administering Oracle Data Safe* guide.
- Grant the SQL Firewall role to the Data Safe service account on the target database. For more information, see Roles for the Oracle Data Safe Service Account in the *Administering Oracle Data Safe* guide.
- Obtain the required IAM permissions which can be granted by a tenancy administrator: To use the full functionality of SQL Firewall it is recommended to be granted `manage` permissions on `data-safe-sql-firewall-family` in the relevant compartments.

  ```
  Allow group <group-name> to manage data-safe-sql-firewall-family in
  compartment
          <compartment-name>
  ```

  Alternatively, administrators may grant more selective permissions by granting permissions to specific resources within `data-safe-sql-firewall-family`. For more information on the resources contained within `data-safe-sql-firewall-family`, see `data-safe-sql-firewall-family` Resource.

2.2.2 Start Using SQL Firewall

In order to begin using SQL Firewall you need to complete the following steps. Ensure you have already completed the prerequisites before starting these steps.

These steps will walk you through

1. Enabling SQL Firewall on your Oracle Database 23ai or above
2. Collecting SQL traffic
3. Stopping the collection of SQL traffic
4. Generating and enforcing SQL Firewall policies
5. Viewing SQL Firewall violation logs
6. Creating audit trails and alert policies for SQL Firewall violations
7. Configuring notifications for SQL Firewall violations

By completing these steps you will be taking steps to protect your database fleet against SQL injection attacks and compromised accounts.

2.2.2.1 Step 1: Enable SQL Firewall On Your Target Database

This steps ensures that SQL Firewall is enabled on your target database.

1. Under **Security center**, click **SQL Firewall**.
2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
   1. (Optional) Select a time period from the **Time period** menu.
   2. (Optional) Select a target database from the **Target database** menu.
4. Click on the name of a target database.
   This will take you to the Configuration details page.
5. Click **Enable**. This can be done through either the button under the name of the database security configuration or the Enable option under the Target database section of the Database security configuration information tab.
6. Wait for the resource to change to Active before continuing to the next step.

2.2.2.2 Step 2: Start SQL Collection for a Database User

This step starts the collection of expected SQL statements and expected database connection paths for the database user. Run the typical application workload from the trusted database connection paths.

1. In the Configuration details page from the previous step, click **Create and start SQL collection.**
2. Select the database user for which collection needs to be created.
3. Select the SQL Collection Level:
   - User issued SQL commands - These are SQL statements that were issued directly from the user to be executed on the database. This is the default.
   - User issued SQL commands and SQL commands issues from PL/SQL units - This includes SQL statements issued directly from the user as well as SQL statements within a PL/SQL unit which is invoked by the user.

   Note: SQL collection will *not* record any internal recursive SQL statements.
4. Click **Create and start SQL collection.**
5. Perform typical daily tasks in your applications for the selected database user.
6. Allow the SQL collection to run for some time. This is discussed further in Step 3: Monitor the Progress of SQL Collection with Insights.

2.2.2.3 Step 3: Monitor the Progress of SQL Collection with Insights

In this step you will monitor the collection of SQL statements and determine when collection can be stopped. Monitor the SQL collection until you see there are no new incoming unique SQL statements or trusted connection paths from the running workload.

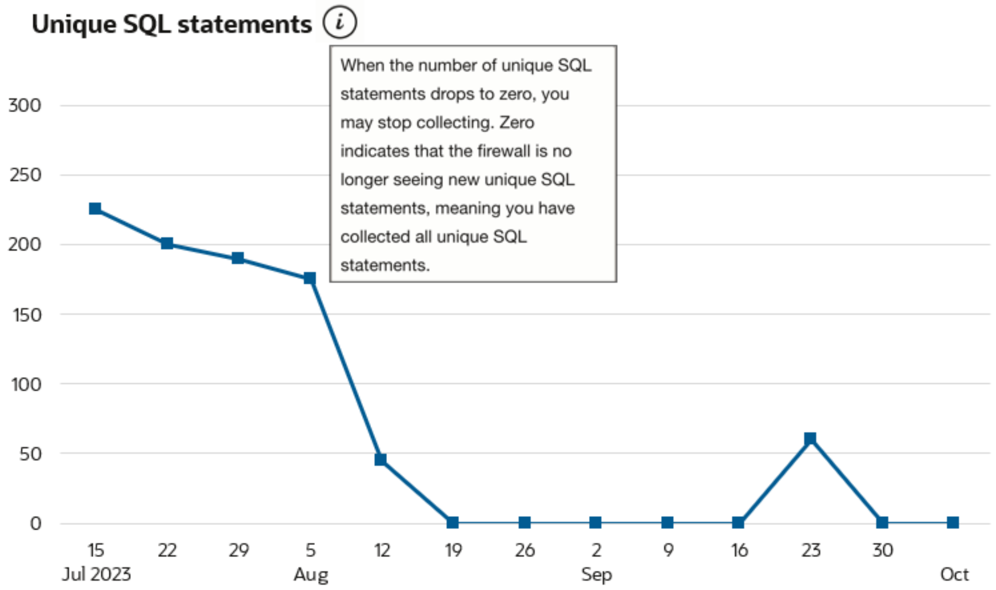
1. Click the **SQL collection insights** tab.

2. The information on the SQL collection tab refreshes every hour, if necessary click **Refresh Insights**.
3. (Optional) Select the time period for which you would like to review the SQL collection.
4. Review the Unique SQL statements chart.
   The collected SQL statements are analyzed to determine if they are unique over the span of the collection period and this chart displays the number of unique SQL statement on the selected time interval. Once there are no more new unique SQL statement being initiated, i.e. the chart remains steady at zero, it is recommended to stop the collection. Waiting until the number of unique SQL statements comes to zero ensures that you collect all statements that are typically executed on your target database and helps establish a status quo.

   For example, if there are 250 SQL statements executed on the first day of the collection but only 225 of those are unique then the chart will show 225 for that day. In the following week if the same 250 statements and an additional 200 new and unique statements are executed then the chart will only show 200. This is because the 250 statements were already collected and observed in week one, thus they are not unique. The number of unique SQL statements will reach zero when there are no more unique SQL statements are observed. See Figure 2-3 for reference.

   It may take several days to weeks for you to collect enough unique SQL statements to stabilize at zero.

Figure 2-3 Unique SQL Statements chart in SQL Collection Insights



When the number of unique SQL statements drops to zero, you may stop collecting. Zero indicates that the firewall is no longer seeing new unique SQL statements, meaning you have collected all unique SQL statements.

5. Review the Client IP, Client OS user, and Client program charts.
   These charts show you the number of client IP addresses, OS users, and programs, respectively, that are executing SQL commands on your target database each day. The specific context information can be viewed in the table below the charts.

   Since SQL statements should be coming from the same session contexts each day, it is recommended to stop the collection when the charts stabilize at a certain value day to day.
6. Review the list of session context types and values.
   Reviewing the list of client IP addresses, client OS users, and client programs allows you to determine where your traffic is coming from. With this information you can set up rules that log or block traffic from all other locations. This is further discussed in Step 4: Generate and Enforce SQL Firewall Policies.

7. Once you have collected a sufficient amount of unique SQL statements, click **Stop** to stop the collection.
   Once you have stopped the collection you will see start time, stop time, and the elapsed time under **Collection timeline** of the **SQL collection information** tab.

2.2.2.4 Step 4: Generate and Enforce SQL Firewall Policies

In this step you will review the information gathered during the collection and create policies with allowlists based on the collected data. Policies will also be enforced to either observe and allow violations or block violations.

1. In the SQL collection details page from the previous step, click **Generate firewall policy.**This will take you to the Firewall policy details page.
2. Review the SQL session context and the Unique allowed SQL statements tables.
   If desired you can add, edit, or remove session context information to be included in the policy but you can't add, edit, or remove any of the collected unique SQL statements in the policy.
3. (Optional) Update the allowed SQL session context values as desired.
   a. Click **Update** for the respective row.
   b. To remove a value, click the **X** at the end of the row in the panel.
   c. To add a value, click **Add** and enter the new value in the empty field.
   d. Click **Update client IP/client program/client OS user**, depending on which context information you selected.
4. (Optional) Download a PDF or XLS report of all Unique allowed SQL statements.
   a. Click **Generate report**.
      A pop-up will appear.
   b. Select which format you want the report in, PDF or XLS.
   c. Enter a name for the report.
   d. Optionally, enter a description for the report.
   e. Click **Generate report**.
   f. Download the report. You have two options:
      - In the Generate report window, click the **here** link. The document will begin downloading.
      - Click **Close** to close the Generate report window. Then, click the **Download report** button. A dialog box is displayed providing you options to open or save the document.
5. Click on **Deploy and Enforce**.
   a. Select the enforcement scope:
      - All (Session contexts and SQL statements)
      - Session contexts only - This option enforces the checks only on the database connection paths.
      - SQL statements only - This option enforces the checks only on the SQL statements.
   b. Select the action on violations:

- Observe (Allow) and log violations - This option will observe and allow all SQL statements and connections to the database while logging any violations.
- Block and log violations - This option will block any SQL statements and database connections not listed in the policy and log the violations. Consider this option when you want SQL Firewall to prevent unauthorized SQL traffic to the database.

c. Audit for violations

   - On - This option will write the violation records to the audit trail. It enables alerting and helps demonstrate compliance to your audit requirements. Ensure to start the audit trail in Data Safe to collect the audit events. These audit events contribute to the monthly free limit of 1 million audit records per month per target database.
   - Off

d. Click **Deploy and enforce**.

2.2.2.5 Step 5: View SQL Firewall Violation Reports

In this step you can view a report of violations for your enforced SQL Firewall policies. There are a variety of ways to navigate to the violations report, some of which will automatically apply filters for your selected SQL Firewall policies, target databases, time periods, and so on.

> **Note:**
>
> It is unlikely that you will see any violations immediately after enforcing a SQL Firewall policy.

- View all violations
- View target specific violations
- View policy specific violations

View all violations

Complete these steps to view a report of all violations across your database fleet.

1. Under **Security center**, click **SQL Firewall.**
2. Under **Related resources**, click **Violation reports.**
3. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
4. Select which report you would like to see from the **Predefined reports** tab:
   - All violations report - Both SQL and context violations
   - SQL violations report - Violations on SQL statements
   - Context violations report - Violations on database connection paths
5. Once at the report you may perform all standard report actions in Oracle Data Safe such as:
   - Apply basic filters

- Apply advanced SCIM filters
- Create custom reports
- Schedule reports
- Generate and download reports
- Manage which columns to display

View target specific violations

Complete these steps to view a report of all violations on a specific target database from the last week.

1. Under **Security center**, click **SQL Firewall**.
2. Click the **Violation summary** tab.
3. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
4. (Optional) Filter the list of results, under **Filters**, do the following:
   a. (Optional) Select a time period from the **Time period** menu.
   b. (Optional) Select a target database from the **Target database** menu.
5. Select the name of a target database from the list.
   This will take you to the violation report.
6. The violation report will be automatically filtered to show only the violations for the selected target database from the selected time period.
7. Once at the report you may perform all standard report actions in Oracle Data Safe such as:
   - Apply basic filters
   - Apply advanced SCIM filters
   - Create custom reports
   - Schedule reports
   - Generate and download reports
   - Manage which columns to display

View policy specific violations

Complete these steps to view a report of violations specific to a selected SQL Firewall policy.

1. Under **Security center**, click **SQL Firewall**.
2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
   a. (Optional) Select a time period from the **Time period** menu.
   b. (Optional) Select a target database from the **Target database** menu.
4. Select the name of a target database from the list on the **Target summary** tab.
   This will take you to the Configuration details page.

5. Under **Resources**, click **SQL Firewall policies**.
6. Select a SQL Firewall policy from the list.
   This will take you to the Firewall policy details page.
7. Under **Enforcement information**, click **View report** next to **Violation reports**.
8. The violation report will be automatically filtered to show only the violations for the selected database user on the selected target database.
   This will take you to the violation report.
9. Once at the report you may perform all standard report actions in Oracle Data Safe such as:
   - Apply basic filters
   - Apply advanced SCIM filters
   - Create custom reports
   - Schedule reports
   - Generate and download reports
   - Manage which columns to display

2.2.2.6 Step 6 (Optional): Create Audit and Alert Policies for SQL Firewall Violations

In this step you will create audit and alerts policies for SQL Firewall violations so that you can better track and monitor activity on your database fleet. Though this step is optional, it is recommended as it enables alerting and helps demonstrate compliance to your audit requirements.

Complete the prerequisites for Activity Auditing and Alerts.

1. Complete the Activity Auditing workflow to audit SQL Firewall violations.
   You need to have turned on **Audit for violations** when enforcing your SQL Firewall policies for the corresponding audit policies to show as enabled in Activity auditing. You can view and manage the audit policies for SQL Firewall listed under the SQL Firewall auditing section of the Audit policy details.

   > **Tip:**
   >
   > You must turn on **Audit for violations** in your SQL Firewall policy before managing the SQL Firewall audit policies in the Activity Auditing workflow. See Update the Enforcement of SQL Firewall Policies for more information.

2. Complete the Alerts workflow to receive alerts for SQL Firewall violations.
   The alert policy for SQL Firewall is **SQL Firewall violations**.

2.2.2.7 Step 7 (Optional): Configure Notifications for SQL Firewall Violations

In this step you will configure notifications for when a SQL Firewall violation occurs. Though this step is optional, it is recommended as it will enable you to receive near real-time alerts in the event of a SQL Firewall violation.

In Data Safe you can create event notifications through a workflow available in SQL Firewall. This allows you to create event notifications in context. You can use the quickstart template for common events or the advanced event notification workflows to create notifications.

Prerequisites:

Ensure you have the necessary IAM permissions to create event notifications. For more information, see Permissions to Use Contextual Event Notifications in the *Administering Oracle Data Safe* guide.

To create notifications:

1. Under **Security center**, click **SQL Firewall**.
2. Click the **Notifications** tab.
3. Click **Create notification**.
   If you don't have any notifications created for the selected resource then you will see a list of available quickstart templates. You may click on one of these instead.

   The **Create notification** side panel will appear.
4. Select to create an event notification from either a **Quickstart** template or an **Advanced event notification**.
   A Quickstart templates allow you to select from a list of common event scenarios. When you create a notification from a quickstart template, the Rule and Event is created automatically.

   > **Note:**
   >
   > The Rule and Event are created in the compartment that you were working in when you started the Notification workflow. Rules and Events will only trigger for the compartment and any child-compartments of the compartment that they were created in.

5. If you selected **Quickstart** in the previous step, make a quickstart **Template selection**.
   If you selected **Advanced event notification** in the previous step, type in a **Rule name** and select an **Event type**.

   See SQL Firewall Event Types in the *Administering Oracle Data Safe* guide for more information on events.
6. Select to either **Create new topic** or to **Select existing topic**.
7. Select a **Compartment**.

   > **Note:**
   >
   > This compartment is where the topic will be created, not where the rule and event will be monitored in.

8. If you're creating a new topic, type the topic name or, if you're using an existing topic, select the topic name.
9. Select a **Subscription protocol**.
10. Provide the necessary inputs for the selected subscription protocol.
11. Optionally, click **Show Advanced Options** to tag the notification.

   a. Click **+ Another Tag** to create an additional optional tag to organize and track resources in your tenancy.
   b. Select a **Tag Namespace** from the drop-down list.
   c. Provide a **Tag Key** and **Tag Value**.
12. Click **Create notification**.

Following the completion of these steps, SQL Firewall will start observing the incoming SQL statements and database connection paths, and will allow or block the SQL traffic to proceed to the target database based on the enforced SQL Firewall policy while logging any violations. You can monitor the SQL Firewall violations in Data Safe. If you configured audit and alert configuration, OCI notifications will be triggered in the event of a SQL Firewall violation.

2.2.3 Gain Insights from SQL Firewall

After successfully setting up SQL Firewall to monitor and block and allow SQL activity on your Oracle Database 23ai target databases, you'll want to ensure that you understand the dashboard and violations report. You should also understand what actions to take in the event of a high volume of violations.

2.2.3.1 View the SQL Firewall Dashboard

When you select **SQL Firewall** under **Security center** in Oracle Data Safe you will see the dashboard of SQL Firewall information for the last week. This dashboard provides you with a high-level view of your SQL Firewall implementation across your fleet of Oracle Database 23ai or above target databases in your selected compartment(s).

To filter the dashboard you can alter the compartments, time period, and databases that you can see information for by:

1. Under **Security center**, click **SQL Firewall**.
2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
   a. (Optional) Select a time period from the **Time period** menu.
   b. (Optional) Select a target database from the **Target database** menu.

The dashboard shows the following information:

- **SQL Firewall violations** chart - Shows the number of SQL statement violations and the number of context violations throughout your Oracle Database 23ai or above fleet per day. This allows you to determine patterns in the number of SQL statement and session context violations and identify spikes in violations that should be investigated.
- **SQL Firewall enforcement mode** chart - Shows you a break down of how many of your SQL Firewall policies either "block" or "observe" SQL statements or session contexts that violate your policies.
- **SQL Collections** chart - Shows you a break down of the number of SQL collections in each life cycle state: `COLLECTING`, `COMPLETED`, `DELETED`, `FAILED`, `NEEDS_ATTENTION`.
- **Target Summary** tab - Shows you a break down per registered Oracle Database 23ai or above of the number of database users that SQL statements are actively being collected for, the number of policies that block violations, and the number of polices that allow and observe violations. You can click on the name of a target database to see its SQL Firewall

configuration details and drill down deeper into the SQL collections, SQL Firewall policies, and Work Requests on the target database.

- **Violations Summary** tab - Shows you a break down per registered Oracle Database 23ai or above of the total number of violations, the number of SQL violations, and the number of Context violations. You can click on the name of a target database to see a more detailed violations report.
- The **Notifications** tab - Shows you what event notifications and subscriptions you have created for SQL Firewall. More specifically, it displays the event, rule name, topic name, and when the event notification was created. This table will only show Events that you have created directly within Data Safe. In addition to displaying existing event notifications, you can also create new notifications by using the Create notification button. See Create and Modify Event Notifications in SQL Firewall for more information.

2.2.3.2 View Violations

There are multiple ways that you can view context and SQL statement violations once you have enforced SQL Firewall policies.

- View all violations
- View target specific violations
- View policy specific violations

View all violations

Complete these steps to view a report of all violations across your database fleet.

1. Under **Security center**, click **SQL Firewall**.
2. Under **Related resources**, click **Violation reports.**
3. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
4. Select which report you would like to see from the **Predefined reports** tab:
   - All violations report - Both SQL and context violations
   - SQL violations report - Violations on SQL statements
   - Context violations report - Violations on database connection paths
5. Once at the report you may perform all standard report actions in Oracle Data Safe such as:
   - Apply basic filters
   - Apply advanced SCIM filters
   - Create custom reports
   - Schedule reports
   - Generate and download reports
   - Manage which columns to display

View target specific violations

Complete these steps to view a report of all violations on a specific target database from the last week.

1. Under **Security center**, click **SQL Firewall**.
2. Click the **Violation summary** tab.
3. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
4. (Optional) Filter the list of results, under **Filters**, do the following:
   a. (Optional) Select a time period from the **Time period** menu.
   b. (Optional) Select a target database from the **Target database** menu.
5. Select the name of a target database from the list.
   This will take you to the violation report.
6. The violation report will be automatically filtered to show only the violations for the selected target database from the selected time period.
7. Once at the report you may perform all standard report actions in Oracle Data Safe such as:
   - Apply basic filters
   - Apply advanced SCIM filters
   - Create custom reports
   - Schedule reports
   - Generate and download reports
   - Manage which columns to display

View policy specific violations

Complete these steps to view a report of violations specific to a selected SQL Firewall policy.

1. Under **Security center**, click **SQL Firewall**.
2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
   a. (Optional) Select a time period from the **Time period** menu.
   b. (Optional) Select a target database from the **Target database** menu.
4. Select the name of a target database from the list on the **Target summary** tab.
   This will take you to the Configuration details page.
5. Under **Resources**, click **SQL Firewall policies**.
6. Select a SQL Firewall policy from the list.
   This will take you to the Firewall policy details page.
7. Under **Enforcement information**, click **View report** next to **Violation reports**.
8. The violation report will be automatically filtered to show only the violations for the selected database user on the selected target database.
   This will take you to the violation report.

9. Once at the report you may perform all standard report actions in Oracle Data Safe such as:
    - Apply basic filters
    - Apply advanced SCIM filters
    - Create custom reports
    - Schedule reports
    - Generate and download reports
    - Manage which columns to display

Related Topics

- View and Manage Violations Report
  Describes actions that can be take on reports and how to create custom reports.

2.2.3.3 Follow-Up Actions for SQL Firewall

In an ideal scenario where the SQL collection has captured all expected SQL statements and trusted database connections, violations indicate potential database attacks such as compromised account access and SQL Injection attacks. But if the collected statements or database connections are not complete or there are new authorized SQL statements following an application update, there is a possibility to see a surge in violations. Ensure to update the SQL Firewall policies to collect these additional statements to avoid false positives in the violation reports.

Related Topics

- Update SQL Firewall Policies

2.2.4 Manage SQL Firewall

Managing your SQL Firewall policies and configurations helps ensure that your databases are protected from threats while also ensuring that intended SQL actions can be taken on your databases. See the below topics for information on how to update your SQL Firewall configurations and policies.

2.2.4.1 Update the Database Security Configuration

1. Under **Security center**, click **SQL Firewall**.
2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
    a. (Optional) Select a time period from the **Time period** menu.
    b. (Optional) Select a target database from the **Target database** menu.
4. Click on the name of a target database.
   This will take you to the Configuration details page.
5. Perform any of the following tasks:

- Click **Disable** next to **SQL Firewall status** to disable SQL Firewall. This will stop any ongoing collections and policies will no longer be enforced.
- Click **Turn on** or **Turn off** next to **Auto-purge violation logs** to turn this on or off. This specifies whether Data Safe should automatically purge the violation logs from the database after collecting the violation logs and persisting them on Data Safe.

> **Note:**
>
> When this is turned on violation logs are automatically purged every seven days.

- Click **Include** or **Exclude** next to **Database jobs** to include or exclude database jobs for SQL Firewall enforcement.
- Click **Refresh** next to **Last refresh time** to refresh Data Safe's copy of the policies if you made a recent policy change within the database.
- Click **Move Resource** to move the Database Security Configuration to a different compartment.

2.2.4.2 Purge a SQL Collection

Purge helps clean the collection logs for the user. You typically need to purge the SQL Collection when you need to recapture an application SQL workload for the same database user following application updates. The SQL collection can be started again for the database user once it is purged.

1. Under **Security center**, click **SQL Firewall**.
2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
   1. (Optional) Select a time period from the **Time period** menu.
   2. (Optional) Select a target database from the **Target database** menu.
4. Click on the name of a target database.
   This will take you to the Configuration details page.
5. Under **Resources**, click **SQL collections**.
6. Click on a database user name.
   This will take you to the SQL collection details page.
7. Click **Purge** to remove the SQL collection. This will not stop any SQL Firewall Policies that were generated from this collection.

2.2.4.3 Drop a SQL Collection

Drop will remove the SQL Collection and collection logs for the selected database user. You typically have to drop the SQL Collection when you need to remove SQL Firewall protection for a database user who is no longer active or has changed responsibilities in the system.

1. Under **Security center**, click **SQL Firewall**.

2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
    a. (Optional) Select a time period from the **Time period** menu.
    b. (Optional) Select a target database from the **Target database** menu.
4. Click on the name of a target database.
   This will take you to the Configuration details page.
5. Under **Resources**, click **SQL Collections**.
6. Click on a database user name.
   This will take you to the SQL collection details page.
7. Click on **More actions** and select **Drop** to delete the SQL collection. Dropping a SQL collection will not have an impact on already generated or enforced SQL Firewall policies.

2.2.4.4 View and Manage SQL Firewall Policies

1. Under **Security center**, click **SQL Firewall**.
2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
    a. (Optional) Select a time period from the **Time period** menu.
    b. (Optional) Select a target database from the **Target database** menu.
4. Click on the name of a target database.
   This will take you to the Configuration details page.
5. Under **Resources**, click **SQL Firewall policies**.
6. Click on a database user name.
   This will take you to the Firewall policy details page.
7. (Optional) Update the allowed SQL session context values as desired.
    a. Click **Update** for the respective row.
    b. To remove a value, click the **X** at the end of the row in the panel.
    c. To add a value, click **Add** and enter the new value in the empty field.
    d. Click **Update client IP/client program/client OS user**, depending on which context information you selected.
8. (Optional) Download a PDF or XLS report of all Unique allowed SQL statements.
    a. Click **Generate report**.
       A pop-up will appear.
    b. Select which format you want the report in, PDF or XLS.
    c. Enter a name for the report.
    d. Optionally, enter a description for the report.
    e. Click **Generate report**.
    f. Download the report. You have two options:

- In the Generate report window, click the **here** link. The document will begin downloading.
- Click **Close** to close the Generate report window. Then, click the **Download report** button. A dialog box is displayed providing you options to open or save the document.

2.2.4.5 Update SQL Firewall Policies

1. Under **Security center**, click **SQL Firewall**.
2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
   a. (Optional) Select a time period from the **Time period** menu.
   b. (Optional) Select a target database from the **Target database** menu.
4. Click on the name of a target database.
   This will take you to the Configuration details page.
5. Under **Resources**, click **SQL collection**.
6. Click on a database user name.
   This will take you to the SQL collections details page.
7. Click on the associated SQL Firewall policy located in the SQL collection information tab. This will take you to the Firewall details page.
8. Temporarily disable the SQL Firewall policy by clicking **Disable**. Confirm disablement in the pop-up by clicking **Disable**.
9. Navigate back to the SQL collection by clicking **SQL collection details** in the page breadcrumbs.
10. Click **Start** to capture SQL statements.
11. Initiate the SQL statements you want to add on your target database.
12. Click **Stop** once you have collected the SQL statements.
13. Click **Update firewall policy** to append the new SQL statements to the associated policy.
14. Click on the associated SQL Firewall policy located in the SQL collection information tab. This will take you to the Firewall details page.
15. Click on **Deploy and Enforce**.
    a. Select the enforcement scope:
       - All (Session contexts and SQL statements)
       - Session contexts only - This option enforces the checks only on the database connection paths.
       - SQL statements only - This option enforces the checks only on the SQL statements.
    b. Select the action on violations:
       - Observe (Allow) and log violations - This option will observe and allow all SQL statements and connections to the database while logging any violations.

- Block and log violations - This option will block any SQL statements and database connections not listed in the policy and log the violations. Consider this option when you want SQL Firewall to prevent unauthorized SQL traffic to the database.

c. Audit for violations

- On - This option will write the violation records to the audit trail. It enables alerting and helps demonstrate compliance to your audit requirements. Ensure to start the audit trail in Data Safe to collect the audit events. These audit events contribute to the monthly free limit of 1 million audit records per month per target database.
- Off

d. Click **Deploy and enforce**.

2.2.4.6 Update the Enforcement of SQL Firewall Policies

1. Under **Security center**, click **SQL Firewall**.
2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
   a. (Optional) Select a time period from the **Time period** menu.
   b. (Optional) Select a target database from the **Target database** menu.
4. Click on the name of a target database.
   This will take you to the Configuration details page.
5. Under **Resources**, click **SQL Firewall policies.**
6. Select a SQL Firewall policy from the list.
   This will take you to the Firewall policy details page.
7. Click on **Deploy and Enforce**.
   a. Select the enforcement scope:
      - All (Session contexts and SQL statements)
      - Session contexts only - This option enforces the checks only on the database connection paths.
      - SQL statements only - This option enforces the checks only on the SQL statements.

   b. Select the action on violations:
      - Observe (Allow) and log violations - This option will observe and allow all SQL statements and connections to the database while logging any violations.
      - Block and log violations - This option will block any SQL statements and database connections not listed in the policy and log the violations. Consider this option when you want SQL Firewall to prevent unauthorized SQL traffic to the database.

   c. Audit for violations
      - On - This option will write the violation records to the audit trail. It enables alerting and helps demonstrate compliance to your audit requirements. Ensure to start the audit trail in Data Safe to collect the audit events. These audit events contribute to the monthly free limit of 1 million audit records per month per target database.
      - Off

   d. Click **Deploy and enforce**.

2.2.4.7 Disable or Enable SQL Firewall Policies

1. Under **Security center**, click **SQL Firewall**.
2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
   a. (Optional) Select a time period from the **Time period** menu.
   b. (Optional) Select a target database from the **Target database** menu.
4. Click on the name of a target database.
   This will take you to the Configuration details page.
5. Under **Resources,** click **SQL Firewall policies.**
6. Select a SQL Firewall policy from the list.
   This will take you to the Firewall policy details page.
7. Click **Disable** or **Enable**. Disabling will stop the SQL Firewall from evaluating any incoming SQL traffic against this SQL Firewall policy. However, this will not delete the policy and it can be enabled again later.

2.2.4.8 Drop SQL Firewall Policies

1. Under **Security center**, click **SQL Firewall**.
2. (Optional) Under **List scope**, select the compartment that contains your target database. Optionally select **Include child compartments** to include target database in the list from child compartments.
3. (Optional) Filter the list of results, under **Filters**, do the following:
   a. (Optional) Select a time period from the **Time period** menu.
   b. (Optional) Select a target database from the **Target database** menu.
4. Click on the name of a target database.
   This will take you to the Configuration details page.
5. Under **Resources**, click **SQL Firewall policies**.
6. Select a SQL Firewall policy from the list.
   This will take you to the Firewall policy details page.
7. Click **Drop**. This will delete the SQL Firewall policy and a SQL Collection will have to be initiated again to re-create this policy.

2.2.5 View and Manage Violations Report

Describes actions that can be take on reports and how to create custom reports.

2.2.5.1 Modifying Columns in a Violations Report

To add or remove columns in the report, do the following:

1. View a predefined or custom violations report.
2. Click on the **Actions** drop down menu.

3. Click **Manage columns**.
   The Manage columns window is displayed.
4. Select columns that you want displayed in the report.
5. Deselect columns that you want to hide in the report.
6. Click **Save changes**.

2.2.5.2 Basic Filtering in a Violations Report

To apply basic filters in the report, do the following:

1. View a custom or predefined violations report.
2. Click **Another filter**.
3. Select a filter type, operator, and enter a value. All columns that are available in the report are available as filter types.
4. Click **Apply**.
5. Repeat steps two through four to apply additional filters.

To remove a filter, click the X beside the filter row.

To filter the report based on a total category (for example, Violations blocked), click the total. The list of violations in the table at the bottom of the report is automatically updated. To remove the filter, click the total again.

> **Note:**
>
> Only some totals in your report are single-click filters.

2.2.5.3 Advanced Filtering in a Violations Report

Advanced filtering of violations can provide flexibility in the way that data is analyzed and reviewed, by allowing organizations to specify complex conditions and multiple criteria that must be met in order for data to be included or excluded from the analysis.

To apply advanced filters in the report, do the following:

1. View a predefined or custom violations report.
2. Click **Show Advanced SCIM Query Builder**.
3. Use the provided filter builder and dropdowns to type in your filter(s). Advanced filtering uses System for Cross-Domain Identity Management (SCIM) syntax and supported operators include:
   - `co`: matches resources with an attribute that contains a given string
   - `eq`: matches resources with an attribute that is equal to a given value (not case sensitive)
   - `eq_cs`: matches resources with an attribute that is equal to a given value (case sensitive)
   - `ew`: matches resources with an attribute that ends with a given string
   - `ge`: matches resources with an attribute that is greater than or equal to a given value
   - `gt`: matches resources with an attribute that is greater than a given value

- `in`: matches resources with an attribute that is equal to any of given values in list
- `le`: matches resources with an attribute that is less than or equal to a given value
- `lt`: matches resources with an attribute that is less than a given value
- `ne`: matches resources with an attribute that is not equal to a given value
- `not_in` : matches resources with an attribute that is not equal to any of given values in list
- `pr`: matches resources with an attribute if it has a given value
- `sw`: matches resources with an attribute that starts with a given string

Operators can be grouped using parentheses to specify the order.

Filters can also be combined using logical operators such as `and` and `or`.

> **Note:**
>
> If you have any basic filters currently applied they will appear in the query builder as well.

4. Click **Apply**.

To clear the query builder, click **Clear**. This will clear any basic filters applied as well.

Example 2-1 Context violations and SQL violations that are allowed advanced filter

```
(violationAction eq "ALLOWED")  and ((violationCause eq "context violation")
or (violationCause  eq "SQL violation"))
```

Example 2-2 SQL violations on a specific target database advanced filter

```
(targetName eq "HRApps") and (violationCause eq "SQL violation")
```

Example 2-3 Actions taken on two specific databases since a specifc time advanced filter

```
(operationTime ge "2023-09-11T00:39:43.295Z") and ((targetName eq "HRApps")
or (targetName eq "TF_AUTOMATION"))
```

2.2.5.4 Tips for Using the Filter Builder to Create Advanced Filters

- Pressing the escape key while in advanced filtering mode will clear the whole query.
- Pressing the space key will display the drop down with the list of available attributes or operators.
- Pressing the space key after entering a value like `targetname (demo_tgt)` will enclose the string with quotes: `("demo_tgt")`.
- Pressing enter will close the drop down listing the operators and attribute names.
- If a value like SQL Firewall policy name has spaces in it, typing space will enclose the first word within quotes, `"policy name"`. You will have to move the cursor back to the enclosed string and continue typing the rest of the string value.

- If you build a filter in advanced filtering that can't be displayed in basic filters, you can't switch back to basic filtering mode. For example, advanced filters with the or condition can't be displayed in basic filtering.
- A custom report with basic filter can be updated with advanced filter and saved.

For more information about SCIM, see the protocol documentation at https://www.rfceditor.org/rfc/rfc7644.

For more information about filtering in SCIM, see the filtering section of the protocol documentation at https://www.rfc-editor.org/rfc/rfc7644#section-3.4.2.2.

2.2.5.5 Create a Custom Violations Report

You can create a custom report from any violations report, including the predefined AllViolations report. The details saved to the custom reports are those that you are currentlyviewing on screen. You may want to create a custom report if you want to preserve thefilters and columns displayed in a report that you are viewing online. You may alsowant to store your custom reports in specific compartments.

1. Under **Security center**, click **SQL Firewall**.
2. Under **Related resources**, click **Violation reports**.
3. Click a report name and modify it as needed. If there aren't any custom reports saved, click the All violations report and make changes to it.
4. Click **Create custom report**.
   The Create custom report dialog box is displayed.
5. Enter a name for your custom report.
6. (Optional) Enter a description for your custom report.
7. Select the compartment to where you want to save your custom report.
8. Click **Create custom report**, and wait for a message that tells you the custom report is created.
9. (Optional) To open and view your custom report, click the **click here** link.
10. (Optional) To return to the report displayed on the screen, click **Close**.

2.2.5.6 Update a Custom Violations Report

1. Under **Security center**, click **SQL Firewall**.
2. Under **Related resources**, click **Violation reports**.
3. Click **Custom reports** tab.
4. Click a custom report name.
5. Modify the report as needed.
6. Click **Save Report**.
   The custom report is updated.

2.2.5.7 Delete a Custom Violations Report

When you delete a custom violations report, the report is permanently deleted and cannot berecovered. You cannot delete the predefined All violations report.

1. Under **Security center**, click **SQL Firewall**.

2. Under **Related resources**, click **Violation reports**.
3. Click **Custom reports** tab.
4. Click a custom report name.
5. Click **Delete report**.

   A Delete report dialog box is displayed, asking you to confirm the deletion.
6. Click **Delete report**.

2.2.5.8 Create or Manage a Schedule for a Violations Report

You can create a schedule for a predefined or custom violation report to generate a PDF or XLS report.

1. Under **Security center**, click **SQL Firewall**.
2. Under **Related resource**, click **Violation reports**.
   The **Violation reports** page is displayed, showing you a list of violation reports.
3. To view a predefined violation report, on the **Predefined reports** tab, in the **Report name** column, click the report name that you want to view.
   The predefined report is displayed.
4. To view a custom violation report, click the **Custom report** tab. In the **Report name** column, click the name of your custom report.
   Your custom report is displayed.
5. Click **Manage report schedule**.
   The **Manage report schedule** panel is displayed, pre-loaded with either the default or modified schedule.
6. (Optional) In the **Schedule report name** box, enter a name for the PDF or XLS report.
7. Select a compartment to store the reports generated by the schedule.
8. For **Report format**, select either a **PDF** or **XLS** output.
9. Select a **Schedule frequency**.
   - If you select weekly, select the day of the week in the **Every** field.
   - If you select monthly, select the day of the month in the **Day** field.
10. In **Time (in UTC)**, select a schedule time.
11. In **Events time span**, select the time span for the violation records.
    For example, selecting Last months and entering 14 pulls violations from the last 14 months from the time the report is run.
12. (Optional) Specify a row limit. If unspecified, the default row limit is 200 rows.
13. Click **Save Schedule**.
    You can access the generated PDF/XLS reports on the **Violation report history** page.

2.2.5.9 View and Manage Violation Report History

The **Violation report history** page lists all the PDF/XLS violations reports that are automatically generated via a schedule or on-demand by users. On this page, you can view the list of reports generated during the past three months, details about those reports, and download reports. Oracle Data Safe stores these reports for up to three months.

1. Under **Security center**, click **SQL Firewall**.

2. Under **Related resources**, click **Violation report history**.
   The Violation report history table is displayed. It contains the following information:
   - **Name** - The name of the violation report
   - **State** - Either **Active** or **Updating**, shows if the report is currently accessible or if it is being updated
   - **Report definition** - Specifies the name of the report that provides data for this scheduled or generated report
   - **Generated time** - The time the report was created
   - **Report type** - Generated or Scheduled. Where generated reports are on-demand reports produced outside of the scheduling system and scheduled reports are those produced by the scheduling system
   - **File format** - PDF or XLS
   - **Download report** - Option to download the report
3. (Optional) Under **Filters**, narrow down the report history page based on the **Report definition**, **Report type**, **File format**, and **Time period**.
4. Click on any report name to see further details including OCID and compartment information.

2.2.6 Create and Modify Event Notifications in SQL Firewall

You can create and modify event notifications in SQL Firewall.

2.2.6.1 Creating Event Notifications for SQL Firewall

In Data Safe you can create event notifications for SQL Firewall related events. You can use the quickstart template for common events or the advanced event notification workflows to create notifications.

Prerequisites:

Ensure you have the necessary IAM permissions to create event notifications. For more information, see Permissions to Use Contextual Event Notifications in the *Administering Oracle Data Safe* guide.

To create notifications:

1. Under **Security center**, click **SQL Firewall**.
2. Click the **Notifications** tab.
3. Click **Create notification**.
   If you don't have any notifications created for the selected resource then you will see a list of available quickstart templates. You may click on one of these instead.

   The **Create notification** side panel will appear.
4. Select to create an event notification from either a **Quickstart** template or an **Advanced event notification**.
   A Quickstart templates allow you to select from a list of common event scenarios. When you create a notification from a quickstart template, the Rule and Event is created automatically.

> **Note:**
>
> The Rule and Event are created in the compartment that you were working in when you started the Notification workflow. Rules and Events will only trigger for the compartment and any child-compartments of the compartment that they were created in.

5. If you selected **Quickstart** in the previous step, make a quickstart **Template selection**. If you selected **Advanced event notification** in the previous step, type in a **Rule name** and select an **Event type**.

   See SQL Firewall Event Types in the *Administering Oracle Data Safe* guide for more information on events.
6. Select to either **Create new topic** or to **Select existing topic**.
7. Select a **Compartment**.

   > **Note:**
   >
   > This compartment is where the topic will be created, not where the rule and event will be monitored in.
8. If you're creating a new topic, type the topic name or, if you're using an existing topic, select the topic name.
9. Select a **Subscription protocol**.
10. Provide the necessary inputs for the selected subscription protocol.
11. Optionally, click **Show Advanced Options** to tag the notification.
    a. Click **+ Another Tag** to create an additional optional tag to organize and track resources in your tenancy.
    b. Select a **Tag Namespace** from the drop-down list.
    c. Provide a **Tag Key** and **Tag Value**.
12. Click **Create notification**.

2.2.6.2 Modifying Event Notifications For SQL Firewall

After creating event notifications in SQL Firewall in Oracle Data Safe, you can modify the notifications you created.

To modify the event and rule:

1. Under **Security center**, click **SQL Firewall**.
2. Click the **Notifications** tab.
3. Click on an existing event from the **Name** column.

   > **Note:**
   >
   > You will only see the Events that were created directly within Data Safe.

This will bring you to the Rule details page which is part of Oracle Cloud Infrastructure (OCI) Events Service. For more information, see the Events section of the OCI Documentation.

To modify the topic and subscription:

1. Under **Security center**, click **SQL Firewall**.
2. Click the **Notifications** tab.
3. Click on an existing topic from the **Topic** column.

> **Note:**
>
> You will only see the Topics that were created directly within Data Safe.

This will bring you to the Topic Details page which is part of Oracle Cloud Infrastructure (OCI) Notifications. For more information, see the Notifications section of the OCI Documentation.