188
DBMS_SQL_FIREWALL

The DBMS_SQL_FIREWALL package enables you to monitor users and detect or prevent SQL injection attacks against those users.

This chapter contains the following topics:

DBMS_SQL_FIREWALL Overview

The DBMS_SQL_FIREWALL PL/SQL package enables you to manage SQL Firewall, which tracks and can block SQL injection attacks.

The DBMS_SQL_FIREWALL package enables you to capture SQL activities of users, create allow-lists (that is, permitted actions) from the captured SQL activities, and then enforce the allow-lists to prevent or detect potential SQL injection attacks. In addition to SQL statements, the allow-list can contain a context list, which is a set of session contexts allowed for database connections. An example of a context can be IP addresses. You can also configure SQL Firewall to not run when Oracle Scheduler is running, because to do so may interfere with Oracle Scheduler operations. After you enable the allow-list, any SQL that the user performs will be monitored by SQL Firewall. SQL that the user performs that is not in the allow-list is considered to be a SQL injection attack. You can configure SQL Firewall to either allow the user to continue performing these SQL operations, or you can block these activities. Note that the SQL operations that violate the allow-list will always be written to a log table that you can query with data dictionary views.

You can configure SQL Firewall in both the root and in individual pluggable databases (PDBs).

Related Topics

- *Oracle Database Oracle SQL Firewall User's Guide*

DBMS_SQL_FIREWALL Security Model

Oracle Database protects the administration of SQL Firewall by storing its metadata in tables in the SYS schema.

Hence, these tables rely on dictionary protection, just as other dictionary tables in SYS do. Therefore, users who have the SELECT ANY TABLE system privilege cannot query these tables unless they also have the SELECT ANY DICTIONARY system privilege or are granted the SELECT object privileges on the tables. Only the SYS user can grant these privileges to other users.

Oracle Database stores the SQL Firewall tables in the SYSAUX tablespace by default. If you want to move the SQL Firewall log tables to a different (user-defined) tablespace, then you must first disable SQL Firewall, and then use the MOVE clause of the ALTER TABLE statement to perform the move operation.

To use the procedures in the DBMS_SQL_FIREWALL package, a user must be granted the SQL_FIREWALL_ADMIN role.

Related Topics

- *Oracle Database Oracle SQL Firewall User's Guide*

DBMS_SQL_FIREWALL Constants

The DBMS_SQL_FIREWALL package provides constants that are used with several SQL Firewall procedures.

These constants are described in the following table.

Table 188-1 DBMS_SQL_FIREWALL Constants

Name	Type	Value	Description
DBMS_SQL_FIREWALL.ENFORCE_ALL	NUMBER	3	Enforces both allowed SQL and allowed contexts when you run the DBMS_SQL_FIREWALL.ENABLE_ALLOW_LIST procedure
DBMS_SQL_FIREWALL.ENFORCE_CONTEXT	NUMBER	1	Enforces allowed contexts when you run the DBMS_SQL_FIREWALL.ENABLE_ALLOW_LIST procedure.
DBMS_SQL_FIREWALL.ENFORCE_SQL	NUMBER	2	Enforces allowed SQL when you run the DBMS_SQL_FIREWALL.ENABLE_ALLOW_LIST procedure
DBMS_SQL_FIREWALL.ALL_LOGS	NUMBER	3	Purges all logs when you run the DBMS_SQL_FIREWALL.PURGE procedure
DBMS_SQL_FIREWALL.CAPTURE_LOG	NUMBER	1	Purges only capture logs when you run the DBMS_SQL_FIREWALL.PURGE procedure
DBMS_SQL_FIREWALL.IP_ADDRESS	NUMBER	3	Specifies the user's IP address when you run the DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT or DBMS_SQL_FIREWALL.DELETE_ALLOWED_CONTEXT procedure
DBMS_SQL_FIREWALL.OS_PROGRAM	NUMBER	1	Specifies the user's operating system program when you run the DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT or DBMS_SQL_FIREWALL.DELETE_ALLOWED_CONTEXT procedure

Table 188-1 (Cont.) DBMS_SQL_FIREWALL Constants

Name	Type	Value	Description
DBMS_SQL_FIREWALL.OS_USERNAME	NUMBER	2	Specifies an operating system name when you run the DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT or DBMS_SQL_FIREWALL.DELETE_ALLOWED_CONTEXT procedure
DBMS_SQL_FIREWALL.SCHEDULER_JOB	NUMBER	1	Indicates whether SQL Firewall will capture and enforce allow-lists for database connections and SQL executions during Oracle Scheduler operations. Use this constant with the DBMS_SQL_FIREWALL.EXCLUDE and DBMS_SQL_FIREWALL.INCLUDE procedures.
DBMS_SQL_FIREWALL.VIOLATION_LOG	NUMBER	2	Purges only violation logs when you run the DBMS_SQL_FIREWALL.PURGE procedure

Summary of DBMS_SQL_FIREWALL Subprograms

This table lists and describes the DBMS_SQL_FIREWALL package subprograms.

Table 188-2 DBMS_SQL_FIREWALL Package Subprograms

Subprogram	Description
ADD_ALLOWED_CONTEXT Procedure	Adds a context to the list of allowed contexts for a user who is configured for SQL Firewall
APPEND_ALLOW_LIST Procedure	Appends additional contents to an existing allow-list by using the existing capture logs or violation logs of the user, or both
APPEND_ALLOW_LIST_SINGLE_SQL Procedure	Appends a single SQL record to the violation log or capture log to an existing allow-list
CREATE_CAPTURE Procedure	Creates a SQL Firewall capture for a specified user at a given level
DELETE_ALLOWED_CONTEXT Procedure	Deletes a SQL Firewall context value that had been assigned to a user
DELETE_ALLOWED_SQL Procedure	Deletes a specified entry from the allowed SQL that had been assigned to a user
DISABLE Procedure	Disables SQL Firewall
DISABLE_ALLOW_LIST Procedure	Disables SQL Firewall allow-list enforcement for a given user
DROP_ALLOW_LIST Procedure	Deletes the SQL Firewall allow-list of a specified user

Table 188-2 (Cont.) DBMS_SQL_FIREWALL Package Subprograms

Subprogram	Description
DROP_CAPTURE Procedure	Drops a SQL Firewall capture and deletes all the associated capture logs
ENABLE Procedure	Enables SQL Firewall
ENABLE_ALLOW_LIST Procedure	Enables SQL Firewall allow-list enforcement for a given user
EXCLUDE Procedure	Prevents SQL Firewall from capturing or enforcing allow-lists for database connections and SQL executions during Oracle Scheduler operations
EXPORT_ALLOW_LIST Procedure	Exports the allow-list of the given user in JSON format, into the CLOB provided from the `allow_list` argument
FLUSH_LOGS Procedure	Flushes all the SQL Firewall logs that reside in the memory into the log tables
GENERATE_ALLOW_LIST Procedure	Generates a SQL Firewall allow-list for the specified user by using data from the existing capture logs of the user
IMPORT_ALLOW_LIST Procedure	Imports the allow-list from the specified `CLOB` for the given user, to the target database
INCLUDE Procedure	Enables SQL Firewall to capture and enforce allow-lists for database connections and SQL executions during Oracle Scheduler operations
PURGE_LOG Procedure	Purges SQL Firewall logs
START_CAPTURE Procedure	Starts a SQL Firewall capture for a user
STOP_CAPTURE Procedure	Stops a SQL Firewall capture for a user
UPDATE_ALLOW_LIST_ENFORCEMENT Procedure	Updates the SQL Firewall allow-list enforcement options for the given user

ADD_ALLOWED_CONTEXT Procedure

This procedure adds a context to the list of allowed contexts for a user's SQL Firewall allow-list.

Syntax

```
DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT (
   username       IN  VARCHAR2,
   context_type   IN  NUMBER,
   value          IN  VARCHAR2);
```

Parameters

Table 188-3 ADD_ALLOWED_CONTEXT Procedure Parameters

Parameter	Description
`username`	Specifies the name of the user who has a SQL Firewall allow-list. To find all the users who has an allow-list, query `DBA_SQL_FIREWALL_ALLOW_LISTS`.

Table 188-3 (Cont.) ADD_ALLOWED_CONTEXT Procedure Parameters

Parameter	Description
context_type	Specifies one of the following context types: • DBMS_SQL_FIREWALL.IP_ADDRESS accepts IPv4 and IPv6 addresses and subnets in the CIDR notation. • 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, such as sqlplus or SQL Developer.
value	Specifies the value of the context_type constant, such as an IP address for DBMS_SQL_FIREWALL.IP_ADDRESS. To allow a local (bequeathed) connection that does not have an IP address, specify with the value Local for the DBMS_SQL_FIREWALL.IP_ADDRESS type. To specify all values of the context (such as all possible operating system programs), then enter the % wild card character.

Usage Notes

- You can find the user's current context type settings by querying the following data dictionary views:
  - DBA_SQL_FIREWALL_ALLOWED_IP_ADDR
  - DBA_SQL_FIREWALL_ALLOWED_OS_PROG
  - DBA_SQL_FIREWALL_ALLOWED_OS_USER
- Before you can add any contexts for the user, the user's allow-list must be created (using the DBMS_SQL_FIREWALL.GENERATE_ALLOW_LIST procedure).
- This procedure can be run when the allow-list is enabled or disabled, and it takes effects immediately.

Example

```
BEGIN
  DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT (
    username       => 'PFITCH',
    context_type   => DBMS_SQL_FIREWALL.OS_PROGRAM,
    value          => 'SQL Developer'
   );
END;
/
```

APPEND_ALLOW_LIST Procedure

This procedure appends additional contents to an existing allow-list by using the existing capture logs or violation logs of the user, or both.

Syntax

```
DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST (
   username       IN  VARCHAR2,
   source         IN  NUMBER);
```

Parameters

Table 188-4 APPEND_ALLOW_LIST Procedure Parameters

Parameter	Description
username	Specifies the name of the user who was designated for the SQL Firewall allow-list. To find this user, query DBA_SQL_FIREWALL_ALLOW_LISTS.
source	Specifies one of the following log types: • DBMS_SQL_FIREWALL.CAPTURE_LOG • DBMS_SQL_FIREWALL.VIOLATION_LOG • DBMS_SQL_FIREWALL.ALL_LOGS

Usage Notes

- DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST processes the specified source logs and identifies contents to be appended to the allow-list. Then it populates the SQL Firewall metadata tables for the allowed SQL and allowed contexts, which will be used during the allow-list enforcement.
- You can run this procedure when the allow-list is either enabled or disabled.
- The change takes effect immediately.
- A new allow-list version number will be associated with all the allowed SQL entries added by the same DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST execution. This new version number will be 1 plus the current maximum allow-list version of the specified user.

Example

```
BEGIN
  DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST (
    username       => 'PFITCH',
    source         => DBMS_SQL_FIREWALL.CAPTURE_LOG
   );
END;
/
```

APPEND_ALLOW_LIST_SINGLE_SQL Procedure

This procedure appends a single SQL record to the violation log or capture log to an existing allow-list.

This procedure is useful for when you want to individually append SQL commands from the violations log or the capture log to an existing allow-list.

Syntax

```
DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST_SINGLE_SQL (
   username       IN  VARCHAR2,
   sql_signature  IN  VARCHAR2,
   current_user   IN  VARCHAR2,
   top_level      IN  VARCHAR2,
   source         IN  NUMBER DEFAULT);
```

Parameters

Table 188-5 APPEND_ALLOW_LIST_SINGLE_SQL Procedure Parameters

Parameter	Description
username	Specifies the name of the user who was designated for the SQL Firewall allow-list. To find this user, query DBA_SQL_FIREWALL_ALLOW_LISTS.
sql_signature	Specifies the signature of the SQL to be added. To find the signature of the SQL for the target record, query the DBA_SQL_FIREWALL_CAPTURE or DBA_SQL_FIREWALL_VIOLATIONS dynamic view.
current_user	Specifies the name of the user who the SQL command was executed as. For example, if user pfitch invokes a definer's rights procedure created in the psmith schema, then all the SQL commands in the procedure are executed as psmith, the current_user. If the procedure is an invoker's rights procedure, then the current_user is the invoker, pfitch.
top_level	Specifies whether the SQL that was executed was top level. Possible values are as follows: • Y (for Yes) means that the target SQL record is top-level (that is, the statement that the user directly runs). • N (for No) means that the target SQL record is not top-level (that is, the SQL command that is issued from PL/SQL units).
source	Specifies the source log to add the SQL record from: • DBMS_SQL_FIREWALL.CAPTURE_LOG • DBMS_SQL_FIREWALL.VIOLATION_LOG (default)

Usage Notes

- DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST_SINGLE_SQL processes the specified source log and identifies the target SQL record to be appended to the allow-list. Then it populates the SQL Firewall metadata tables for the allowed SQL, which will be used during the allow-list enforcement.
- You can run this procedure when the allow-list is either enabled or disabled.
- The change takes effect immediately.
- A new allow-list version number will be associated with the newly added allowed SQL entry.

Example

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. Obtain the values for the USERNAME, SQL_SIGNATURE, CURRENT_USER, and TOP_LEVEL columns for the target SQL record.
2. Enter these values in the DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST_SINGLE_SQL SQL procedure to add the target SQL record to the allow-list.
   For example:

```
BEGIN
  DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST_SINGLE_SQL (
    username       => 'PFITCH',
    sql_signature  =>
'7D33A84D0A1B56E382B9A92D01BCD19933969CB16E2AB4934A2258563F5ADB44',
```

```
      current_user   => 'PSMITH',
      top_level      => 'N',
      source         => DBMS_SQL_FIREWALL.CAPTURE_LOG
     );
  END;
  /
```

CREATE_CAPTURE Procedure

This procedure creates a SQL Firewall capture for a specified user at a given level.

Syntax

```
DBMS_SQL_FIREWALL.CREATE_CAPTURE (
   username       IN VARCHAR2,
   top_level_only IN BOOLEAN,
   start_capture  IN BOOLEAN);
```

Parameters

Table 188-6 CREATE_CAPTURE Procedure Parameters

Parameter	Description
username	Specifies the name of the user whose SQL Firewall capture is to be created. To find existing users, query DBA_SQL_FIREWALL_CAPTURES.
top_level_only	• TRUE captures only SQL statements that have been directly issued by the user • FALSE captures both top-level SQL statements and SQL statements that have been issued by PL/SQL units. This setting is the default.
start_capture	• TRUE starts the capture process right away, after you run DBMS_SQL_FIREWALL.CREATE_CAPTURE. This setting is the default. • FALSE does not start the capture process. You can start it later on by using DBMS_SQL_FIREWALL.START_CAPTURE.

Usage Notes

To find the status of existing SQL Firewall captures, including users who have already been configured for SQL Firewall captures, query the DBA_SQL_FIREWALL_CAPTURES data dictionary view.

Example

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

DELETE_ALLOWED_CONTEXT Procedure

This procedure deletes a context from the list of allowed contexts for a user's SQL Firewall allow-list.

Syntax

```
DBMS_SQL_FIREWALL.DELETE_ALLOWED_CONTEXT (
   username       IN  VARCHAR2,
   context_type   IN  NUMBER,
   value          IN  VARCHAR2);
```

Parameters

Table 188-7 DELETE_ALLOWED_CONTEXT Procedure Parameters

Parameter	Description
username	Specifies the name of the user who was designated for the SQL Firewall allow-list. To find this user, query DBA_SQL_FIREWALL_ALLOW_LISTS.
context_type	• DBMS_SQL_FIREWALL.IP_ADDRESS accepts IPv4 and IPv6 addresses and subnets in the CIDR notation. • 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, such as sqlplus or SQL Developer.
value	Specifies the value of the context_type constant, such as an IP address for DBMS_SQL_FIREWALL.IP_ADDRESS. If you omit this value or specify NULL, then all the allowed context values of the specified context type are deleted. This setting is the default.

Usage Notes

- You can find the user's current context type settings by querying the following data dictionary views:
  - DBA_SQL_FIREWALL_ALLOWED_IP_ADDR
  - DBA_SQL_FIREWALL_ALLOWED_OS_PROG
  - DBA_SQL_FIREWALL_ALLOWED_OS_USER
- This procedure can be run when the allow-list is enabled or disabled, and it takes effects immediately.

Example

```
BEGIN
  DBMS_SQL_FIREWALL.DELETE_ALLOWED_CONTEXT, (
    username       => 'PFITCH',
    context_type   => DBMS_SQL_FIREWALL.OS_PROGRAM,
    value          => 'SQL Developer'
   );
END;
/
```

DELETE_ALLOWED_SQL Procedure

This procedure deletes a specified entry from the list of allowed SQL for a user's SQL Firewall allow-list

Syntax

```
DBMS_SQL_FIREWALL.DELETE_ALLOWED_SQL (
   username       IN  VARCHAR2,
   allowed_sql_id IN  NUMBER);
```

Parameters

Table 188-8 DELETE_ALLOWED_SQL Procedure Parameters

Parameter	Description
`username`	Specifies the name of the user who was designated for the SQL Firewall allow-list. To find this user, query `DBA_SQL_FIREWALL_ALLOW_LISTS`.
`allowed_sql_id`	Specifies the ID of the allowed SQL entry to be deleted from the allowed SQL of this user.To find this value, query `DBA_SQL_FIREWALL_ALLOWED_SQL`.

Usage Notes

- You can run this procedure when the allow-list is either enabled or disabled.
- The change takes effect immediately.

Example

```
BEGIN
  DBMS_SQL_FIREWALL.DELETE_ALLOWED_SQL (
    username         => 'PFITCH',
    allowed_sql_id   => 1
   );
END;
/
```

DISABLE Procedure

This procedure disables SQL Firewall and stops all the existing captures and allow-lists that are enabled.

Syntax

```
DBMS_SQL_FIREWALL.DISABLE;
```

Parameters

None

Usage Notes

You can find the current status of SQL Firewall by querying the `DBA_SQL_FIREWALL_STATUS` data dictionary view.

Example

```
EXEC DBMS_SQL_FIREWALL.DISABLE;
```

DISABLE_ALLOW_LIST Procedure

This procedure immediately disables SQL Firewall allow-list enforcement for a given user.

Syntax

```
DBMS_SQL_FIREWALL.DISABLE_ALLOW_LIST (
   username       IN  VARCHAR2);
```

Parameters

Table 188-9 DISABLE_ALLOW_LIST Procedure Parameters

Parameter	Description
username	Specifies the name of the user who was designated for the SQL Firewall allow-list. To find this user, query DBA_SQL_FIREWALL_ALLOW_LISTS. If you specify NULL, then all allow-lists that are currently enabled will be disabled.

Usage Notes

To find the status of users' allow-lists, query the DBA_SQL_FIREWALL_ALLOW_LISTS data dictionary view.

Example

```
EXEC DBMS_SQL_FIREWALL.DISABLE_ALLOW_LIST ('PFITCH');
```

DROP_ALLOW_LIST Procedure

This procedure deletes the SQL Firewall allow-list of a specified user.

Syntax

```
DBMS_SQL_FIREWALL.DROP_ALLOW_LIST (
   username       IN  VARCHAR2);
```

Parameters

Table 188-10 DROP_ALLOW_LIST Procedure Parameters

Parameter	Description
username	Specifies the name of the user who was designated for the SQL Firewall allow-list. To find this user, query DBA_SQL_FIREWALL_ALLOW_LISTS.

Usage Notes

- To find the status of users' allow-lists, query the DBA_SQL_FIREWALL_ALLOW_LISTS data dictionary view.

- You cannot drop an allow-list that is currently enabled. To disable an allow-list, run the `DBMS_SQL_FIREWALL.DISABLE_ALLOW_LIST` procedure.

Example

```
EXEC DBMS_SQL_FIREWALL.DROP_ALLOW_LIST ('PFITCH');
```

DROP_CAPTURE Procedure

This procedure drops a SQL Firewall capture and deletes all the associated capture logs.

Syntax

```
DBMS_SQL_FIREWALL.DROP_CAPTURE (
   username       IN VARCHAR2);
```

Parameters

Table 188-11 DROP_CAPTURE procedure Parameters

Parameter	Description
`username`	Specifies the name of the user whose SQL Firewall capture is to be dropped. To find this user, query `DBA_SQL_FIREWALL_CAPTURES`.

Usage Notes

- To find the status of existing SQL Firewall captures, query the `DBA_SQL_FIREWALL_CAPTURES` data dictionary view.
- You cannot drop a capture that is currently running. To stop the capture, run the `DBMS_SQL_FIREWALL.STOP_CAPTURE` procedure.
- Dropping a capture for a user does not affect the user's allow-list, which can continue to run even if the capture has been dropped. Captures and allow-lists are separate entities.

Example

```
EXEC DBMS_SQL_FIREWALL.DROP_CAPTURE ('C##HR_ADMIN');
```

ENABLE Procedure

This procedure enables SQL Firewall and starts all existing captures and allow-lists that are configured to be enabled.

Syntax

```
DBMS_SQL_FIREWALL.ENABLE;
```

Parameters

None

Usage Notes

You can find the current status of SQL Firewall by querying the `DBA_SQL_FIREWALL_STATUS` data dictionary view.

Example

```
EXEC DBMS_SQL_FIREWALL.ENABLE;
```

ENABLE_ALLOW_LIST Procedure

This procedure immediately enables SQL Firewall allow-list enforcement for a given user.

Syntax

```
DBMS_SQL_FIREWALL.ENABLE_ALLOW_LIST (
   username       IN  VARCHAR2,
   enforce        IN  NUMBER,
   block          IN  BOOLEAN;
```

Parameters

Table 188-12 ENABLE_ALLOW_LIST Procedure Parameters

Parameter	Description
`username`	Specifies the name of the user whose SQL Firewall allow-list is to be enabled. To find this user, query `DBA_SQL_FIREWALL_ALLOW_LISTS`. If you enter `NULL`, then the allow-lists for all users who do not yet have allow-lists enabled are enabled.
`enforce`	• `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`	• `TRUE` blocks 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.

Usage Notes

- To find the status of users' allow-lists, query the `DBA_SQL_FIREWALL_ALLOW_LISTS` data dictionary view.
- SQL Firewall always generates a violation log for any unmatched database connection or SQL statement regardless of the `block` option setting.

Example

```
BEGIN
  DBMS_SQL_FIREWALL.ENABLE_ALLOW_LIST (
    username       => 'PFITCH',
    enforce        => DBMS_SQL_FIREWALL.ENFORCE_SQL,
```

```
    block          => TRUE
   );
END;
/
```

EXCLUDE Procedure

This procedure prevents SQL Firewall from capturing or enforcing allow-lists for database connections and SQL executions during Oracle Scheduler operations.

Oracle Scheduler jobs are often used in databases for various maintenance purposes. Accidentally interrupting critical jobs can cause undesirable consequences. You can configure SQL Firewall to not capture any SQL statements nor enforce any allow-lists that are run during an Oracle Scheduler job session. This procedure applies to all users that have been configured for SQL Firewall captures and allow-lists. By default, Oracle Scheduler jobs are excluded from SQL Firewall operations.

Syntax

```
DBMS_SQL_FIREWALL.EXCLUDE (
   FEATURE        IN NUMBER);
```

Parameters

Table 188-13 EXCLUDE Procedure Parameters

Parameter	Description
FEATURE	Enter DBMS_SQL_FIREWALL.SCHEDULER_JOB for this value.

Usage Notes

- To find the status of whether SQL Firewall is enforced during Oracle Scheduler operations, query the EXCLUDE_JOBS column of the DBA_SQL_FIREWALL_STATUS data dictionary view. If the output is Y, then Oracle Scheduler jobs are excluded from SQL Firewall operations.
- To enable Oracle Firewall to run during Oracle Scheduler operations, run the DBMS_SQL_FIREWALL.INCLUDE procedure.

Example

```
EXEC DBMS_SQL_FIREWALL.EXCLUDE (DBMS_SQL_FIREWALL.SCHEDULER_JOB);
```

EXPORT_ALLOW_LIST Procedure

This procedure exports the allow-list of the given user in JSON format, into the CLOB provided from the allow_list argument.

Syntax

```
DBMS_SQL_FIREWALL.EXPORT_ALLOW_LIST (
   username       IN      VARCHAR2,
   allow_list     IN/OUT  CLOB;
```

Parameters

Table 188-14 EXPORT_ALLOW_LIST Procedure Parameters

Parameter	Description
username	Specifies the user that the allow-list was created for. To find which user has an allow-list, query DBA_SQL_FIREWALL_ALLOW_LISTS.
allow_list	Specifies the CLOB (which must already exist) into which the exported allow-list must go

Usage Notes

- Before you run this procedure, you must create the CLOB and then pass it to the API (for example, by DBMS_LOB.CREATETEMPORARY for the PL/SQL client, or by OracleConnection.createClob() for JDBC Java client).
- The export operation includes the allow-list's settings (status, enforce, block, top_level_only, generated_on, and status_updated_on timestamp), allowed SQL, and allowed contexts. In addition, the export operation includes all the referenced SQL logs (by the allowed SQL).
- DBMS_SQL_FIREWALL.EXPORT_ALLOW_LIST does not export capture logs or violation logs.
- To find the status of users' allow-lists, query the DBA_SQL_FIREWALL_ALLOW_LISTS data dictionary view.
- If you want to export all the SQL Firewall metadata, which includes captures and allow-lists for all users, then instead of using DBMS_SQL_FIREWALL.EXPORT_ALLOW_LIST, use the include=SQL_FIREWALL clause in the Oracle Data Pump expdp command. See *Oracle Database Security Guide*.

Example

```
BEGIN
  DBMS_SQL_FIREWALL.EXPORT_ALLOW_LIST (
    username       => 'PFITCH',
    allow_list     => ALLOW_LIST_CLOB;
   );
END;
/
```

FLUSH_LOGS Procedure

This procedure flushes all the SQL Firewall logs that reside in the memory into the log tables.

Syntax

```
DBMS_SQL_FIREWALL.FLUSH_LOGS;
```

Parameters

None

Usage Notes

- Usually you do not need to invoke this procedure explicitly, because logs in the memory are flushed to the log tables frequently in the background. But in case if you want to see the capture logs or violation logs immediately after the action during when SQL Firewall is running, you can run this procedure before looking at the logs.
- The `DBMS_SQL_FIREWALL.FLUSH_LOGS` procedure is equivalent to the `DBMS_MEMOPTIMIZE_ADMIN.WRITES_FLUSH` procedure. (See WRITES_FLUSH Procedure.)

Example

```
EXEC DBMS_SQL_FIREWALL.FLUSH_LOGS;
```

GENERATE_ALLOW_LIST Procedure

This procedure generates a SQL Firewall allow-list for the specified user by using the existing capture logs of the user.

Syntax

```
DBMS_SQL_FIREWALL.GENERATE_ALLOW_LIST (
   username       IN  VARCHAR2;
```

Parameters

Table 188-15 GENERATE_ALLOW_LIST Procedure Parameters

Parameter	Description
`username`	Specifies the name of the user who was designated for the SQL Firewall allow-list. To find this user, query `DBA_SQL_FIREWALL_CAPTURES`.

Usage Notes

- To find information about existing generated allow-lists, query the `DBA_SQL_FIREWALL_ALLOW_LISTS` data dictionary view.
- Before you run this procedure, the following components must be in place:
  - The specified user must exist.
  - A capture (using `DBMS_SQL_FIREWALL.CREATE_CAPTURE`) has been created for this user. This capture must be disabled (using `DBMS_SQL_FIREWALL.STOP_CAPTURE`) before you can generate an allow-list for the user.
  - No allow-list exists yet for the user.

Example

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

IMPORT_ALLOW_LIST Procedure

This procedure imports the allow-list from the specified `CLOB` for the given user, to the target database.

Syntax

```
DBMS_SQL_FIREWALL.IMPORT_ALLOW_LIST (
   username       IN      VARCHAR2,
   allow_list     IN      CLOB;
```

Parameters

Table 188-16 IMPORT_ALLOW_LIST Procedure Parameters

Parameter	Description
`username`	Specifies the user of the exported allow-list. To check whether this user already had an allow-list created in the target database, query `DBA_SQL_FIREWALL_ALLOW_LISTS`.
`allow_list`	Specifies the `CLOB` that was created when the allow-list was exported with `DBMS_SQL_FIREWALL.EXPORT_ALLOW_LIST`.

Usage Notes

- If this user does not have an allow-list in the target database, a new allow-list will be created for this user using the allow-list from the JSON payload. The new allow-list will have the same settings (`status`, `top_level_only`, `enforce`, `block`, `generated_on`, `status_updated_on`), same allowed contexts and same allowed SQL as the one in the JSON. If the specified user already has an allow-list in the target database, then all the settings (`status`, `top_level_only`, `enforce`, `block`, and various timestamps) of the existing allow-list will remain untouched, but only the allowed SQL and allowed contexts from the JSON will be merged into the ones for the existing allow-list.
- In addition, the import operation includes all the referenced SQL logs (by the allowed SQL).
- To find the status of users' allow-lists, query the `DBA_SQL_FIREWALL_ALLOW_LISTS` data dictionary view.
- If you want to import all the SQL Firewall metadata, which includes captures and allow-lists, then instead of using `DBMS_SQL_FIREWALL.IMPORT_ALLOW_LIST`, use the `include=SQL_FIREWALL` clause in the Oracle Data Pump `impdp` command. See *Oracle Database Security Guide*.

Example

```
BEGIN
  DBMS_SQL_FIREWALL.IMPORT_ALLOW_LIST (
    username       => 'PFITCH',
    allow_list     => ALLOW_LIST_CLOB;
   );
END;
/
```

INCLUDE Procedure

This procedure enables SQL Firewall to capture and enforce allow-lists for database connections and SQL executions during Oracle Scheduler operations.

Syntax

```
DBMS_SQL_FIREWALL.INCLUDE (
   FEATURE        IN NUMBER);
```

Parameters

Table 188-17 INCLUDE Procedure Parameters

Parameter	Description
FEATURE	Enter DBMS_SQL_FIREWALL.SCHEDULER_JOB for this value.

Usage Notes

- To find the status of whether SQL Firewall is enforced during Oracle Scheduler operations, query the EXCLUDE_JOBS column of the DBA_SQL_FIREWALL_STATUS data dictionary view. If the output is N, then SQL Firewall can perform during Oracle Scheduler operations.
- To prevent SQL Firewall from running during Oracle Scheduler operations, run the DBMS_SQL_FIREWALL.EXCLUDE procedure.

Example

```
EXEC DBMS_SQL_FIREWALL.INCLUDE (DBMS_SQL_FIREWALL.SCHEDULER_JOB);
```

PURGE_LOG Procedure

This procedure purges SQL Firewall logs that belong to the given user based on the specified purge time (that is, logs that were generated before the specified purge time).

Syntax

```
BEGIN
  DBMS_SQL_FIREWALL.PURGE_LOG (
   username       IN  VARCHAR2,
   purge_time     IN  TIMESTAMP WITH TIME ZONE,
   log_type       IN  NUMBER);
```

Parameters

Table 188-18 PURGE_LOG Procedure Parameters

Parameter	Description
username	Specifies the user whose capture logs or violation logs you want to purge. To see capture logs, query DBA_SQL_FIREWALL_CAPTURE_LOGS; to see violation logs, query DBA_SQL_FIREWALL_VIOLATIONS.

Table 188-18 (Cont.) PURGE_LOG Procedure Parameters

Parameter	Description
purge_time	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	Specifies the type of the logs to be purged. • DBMS_SQL_FIREWALL.CAPTURE_LOG • DBMS_SQL_FIREWALL.VIOLATION_LOG • DBMS_SQL_FIREWALL.ALL_LOGS (default)

Usage Notes

To find information about SQL Firewall logs, query the DBA_SQL_FIREWALL_VIOLATIONS data dictionary view.

Example

```
BEGIN
  DBMS_SQL_FIREWALL.PURGE_LOG (
    username     => 'PSMITH',
    purge_time   => TO_TIMESTAMP_TZ('23-JAN-22 18.44.42 -07:00', 'DD/MM/YY
HH24:MI:SS TZH:TZM'),
    log_type     => DBMS_SQL_FIREWALL.VIOLATION_LOG
   );
END;
/
```

START_CAPTURE Procedure

This procedure immediately starts a SQL Firewall capture for a user.

Syntax

```
DBMS_SQL_FIREWALL.START_CAPTURE (
   username       IN  VARCHAR2);
```

Parameters

Table 188-19 START_CAPTURE Procedure Parameters

Parameter	Description
username	Specifies the name of the user to be designated for the SQL Firewall capture.

Usage Notes

- A user can only have one SQL Firewall capture. To find if the user already has been configured for a capture, query the DBA_SQL_FIREWALL_CAPTURES data dictionary view.
- After you start the capture process, all SQL the user enters is captured into the SQL Firewall capture log table. You can periodically check the this SQL by querying the DBA_SQL_FIREWALL_CAPTURE_LOGS data dictionary view.

Example

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

STOP_CAPTURE Procedure

This procedure immediately stops a SQL Firewall capture for a given user.

Syntax

```
DBMS_SQL_FIREWALL.STOP_CAPTURE (
   username       IN  VARCHAR2);
```

Parameters

Table 188-20 STOP_CAPTURE Procedure Parameters

Parameter	Description
username	Specifies the name of the user who was designated for the SQL Firewall capture. To find this user, query DBA_SQL_FIREWALL_CAPTURES.

Usage Notes

- The capture process must be currently running before you can run this procedure. You can check its status by querying the DBA_SQL_FIREWALL_CAPTURES data dictionary view.
- After you stop the capture process, you can generate an allow-list for the user by running the DBMS_SQL_FIREWALL.GENERATE_ALLOW_LIST procedure.

Example

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

UPDATE_ALLOW_LIST_ENFORCEMENT Procedure

This procedure immediately updates the SQL Firewall allow-list enforcement options for the given user.

Syntax

```
BEGIN
  DBMS_SQL_FIREWALL.UPDATE_ALLOW_LIST_ENFORCEMENT (
   username       IN  VARCHAR2,
   enforce        IN  NUMBER,
   block          IN  BOOLEAN);
```

Parameters

Table 188-21 UPDATE_ALLOW_LIST_ENFORCEMENT Procedure Parameters

Parameter	Description
username	Specifies the name of the user for whom the allow-list was generated. To find this user, query DBA_SQL_FIREWALL_ALLOW_LISTS. If you enter NULL, then the enforcement options of all the existing allow-lists (both enabled or disabled allow-lists) are updated.
enforce	• 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	• TRUE blocks 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.

Usage Notes

To find the status of users' allow-lists, query the DBA_SQL_FIREWALL_ALLOW_LISTS data dictionary view.

Example

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