## NAME

selabel_db − userspace SELinux labeling interface: DB objects contexts backend.

## SYNOPSIS

**#include <selinux/selinux.h>**

**#include <selinux/label.h>**

**int selabel_lookup(struct selabel_handle ****hnd***,**
**security_context_t ****context***,**

**const char ****object_name***, int** *object_type***);**

## DESCRIPTION

The DB contexts backend maps from a pair of object name and class into security contexts. It is used to find the appropriate context for database objects when relabeling a certain database.

The *object_name* should be fully qualified name using the hierarchy of database objects. For example, the **pg_class** table in the **postgres** database and **pg_catalog** schema should be qualified as **postgres.pg_catalog.pg_class .**

The *object_type* argument should be set to one of the following values:

**SELABEL_DB_DATABASE**
The *object_name* argument specifies the name of a database itself, such as "postgres".

**SELABEL_DB_SCHEMA**
The *object_name* argument specifies the name of a schema object, such as "postgres.public".

**SELABEL_DB_TABLE**
The *object_name* argument specifies the name of a table object, such as "postgres.public.my_table"

**SELABEL_DB_COLUMN**
The *object_name* argument specifies the name of a column object, such as "postgres.public.my_table.user_id"

**SELABEL_DB_TUPLE**
The *object_name* argument specifies the name of a table object which contains the tuples to be relabeled, such as "postgresql.public.my_table". Note that we have no way to identify individual tuple objects, except for WHERE clause on DML statements, because it has no name.

**SELABEL_DB_PROCEDURE**
The *object_name* argument specifies the name of a procedure object, such as "postgres.public.my_func". Note that we don't support to lookup individual security contexts for each procedures which have same name but different arguments.

**SELABEL_DB_SEQUENCE**
The *object_name* argument specifies the name of a sequence object, such as "postgres.public.my_seq".

**SELABEL_DB_BLOB**
The *object_name* argument specifies the name of a large object, such as "postgres.16308". Note that a large object does not have its name, so it is identified by its identifier value.

## OPTIONS

In addition to the global options described in **selabel_open**(3), this backend recognizes the following options:

**SELABEL_OPT_PATH**

A non-null value for this option specifies a path to a file that will be opened in lieu of the standard DB contexts file. It tries to open the specfile designed for SE-PostgreSQL in the default, so if another RDBMS uses this interface, it needs to give an explicit specfile designed for the RDBMS.

## SEE ALSO

**selabel_open**(3), **selabel_lookup**(3), **selabel_stats**(3), **selinux**(8)