
*Full credit is given to the above companies including the OS that this PDF file was generated!*

## *Red Hat Enterprise Linux Release 9.2 Manual Pages on 'x_contexts.5' command*

***$ man x_contexts.5***

selabel_x(5) SELinux API documentation selabel_x(5)

NAME

selabel_x - userspace SELinux labeling interface and configuration file format for the X Window System contexts backend. This backend is also used to determine the default context for labeling remotely connected X clients

SYNOPSIS

```
#include <selinux/label.h>
int selabel_lookup(struct selabel_handle *hnd,
                   char **context,
                   const char *object_name, int object_type);
int selabel_lookup_raw(struct selabel_handle *hnd,
                   char **context,
                   const char *object_name, int object_type);
```

DESCRIPTION

The X contexts backend maps from X Window System object names into se? curity contexts. It is used to find the appropriate context for X Win? dow System objects whose significance and/or usage semantics are deter? mined primarily by name. The returned context must be freed using freecon(3).

selabel_lookup(3) describes the function with its return and error codes.

This backend is also used to determine the default context for labeling

remotely connected X clients.

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

SELABEL_X_PROP

The object_name argument specifies the name of a window property, such as "WM_NAME".

SELABEL_X_SELN

The object_name argument specifies the name of a selec? tion, such as "PRIMARY".

SELABEL_X_EXT

The object_name argument specifies the name of a protocol extension, such as "RENDER".

SELABEL_X_EVENT

The object_name argument specifies the name of an event type, such as "X11:ButtonPress".

SELABEL_X_CLIENT

The object_name argument is ignored, however it should be set to either * (an asterisk or 'wildcard' that will se? lect the default entry) or a specific entry such as "re? mote" in the X contexts file as shown in the EXAMPLE sec? tion. The default context for labeling remote X clients is then returned.

SELABEL_X_POLYPROP

Like SELABEL_X_PROP, but checks if the property was marked as being polyinstantiated. See NOTES below.

SELABEL_X_POLYSELN

Like SELABEL_X_SELN, but checks if the selection was marked as being polyinstantiated. See NOTES below.

Any messages generated by selabel_lookup(3) are sent to stderr by de? fault, although this can be changed by selinux_set_callback(3).

selabel_lookup_raw behaves identically to selabel_lookup but does not perform context translation.

The FILES section details the configuration files used to determine the X object context.

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 X con? texts file (see the FILES section for details).

FILES

The X context file used to retrieve a default context depends on the SELABEL_OPT_PATH parameter passed to selabel_open(3). If NULL, then the SELABEL_OPT_PATH value will default to the active policy X contexts lo? cation (as returned by selinux_x_context_path(3)), otherwise the actual SELABEL_OPT_PATH value specified is used.

The default X object contexts file is:

/etc/selinux/{SELINUXTYPE}/contexts/x_contexts

Where {SELINUXTYPE} is the entry from the selinux configuration file config (see selinux_config(5)).

The entries within the X contexts file are shown in the Object Name String Values and FILE FORMAT sections.

Object Name String Values

The string name assigned to each object_type argument that can be present in the X contexts file are:

| object_type | Text Name |
| --- | --- |
| SELABEL_X_PROP | property |
| SELABEL_X_SELN | selection |
| SELABEL_X_EXT | extension |
| SELABEL_X_EVENT | event |

?SELABEL_X_CLIENT ? client ?

??????????????????????????????????????

?SELABEL_X_POLYPROP ? poly_property ?

??????????????????????????????????????

?SELABEL_X_POLYSELN ? poly_selection ?

??????????????????????????????????????

FILE FORMAT

Each line within the X contexts file is as follows:

object_type object_name context

Where:

object_type

This is the string representation of the object type shown in the Object Name String Values section. There can be multiple lines with the same object_type string that will form a block of entries (each with a different object_name entry).

object_name

These are the object names of the specific X-server re? source such as PRIMARY, CUT_BUFFER0 etc. They are gener? ally defined in the X-server source code (protocol.txt and BuiltInAtoms in the dix directory of the xorg-server source package). The entry can contain '*' for wildcard matching or '?' for substitution. Note that if the '*' is used, then be aware that the order of entries in the file is important. The '*' on its own is used to ensure a default fallback context is assigned and should be the last entry in the object_type block.

context

The security context that will be applied to the object.

Example 1:

```
# object_type  object_name  context
selection      PRIMARY      system_u:object_r:clipboard_xselection_t:s0
selection      *            system_u:object_r:xselection_t:s0
```

Example 2 - This example shows how a client entry can be configured to ensure an entry is always found:

```
# object_type  object_name  context
client         *            system_u:object_r:remote_t:s0
```

NOTES

1. Properties and selections are marked as either polyinstantiated or not. For these name types, the "POLY" option searches only the names marked as being polyinstantiated, while the other option searches only the names marked as not being polyinstantiated. Users of the interface should check both mappings, optionally taking ac? tion based on the result (e.g. polyinstantiating the object).
2. If contexts are to be validated, then the global option SELA? BEL_OPT_VALIDATE must be set before calling selabel_open(3). If this is not set, then it is possible for an invalid context to be returned.

SEE ALSO

selinux(8), selabel_open(3), selabel_lookup(3), selabel_stats(3), selabel_close(3), selinux_set_callback(3), selinux_x_context_path(3), freecon(3), selinux_config(5)

Security Enhanced Linux 29 Nov 2011 selabel_x(5)