

**NAME**

selabel\_x – userspace SELinux labeling interface: X Window System 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 X contexts backend maps from X Window System object names into security contexts. It is used to find the appropriate context for X Window System objects whose significance and/or usage semantics are determined primarily by name. 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 selection, 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 and should be set to NULL. The default context for labeling remote X clients is 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.

**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 contexts file.

**NOTES**

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 action based on the result (e.g. polyinstantiating the object).

**SEE ALSO****selabel\_open(3), selabel\_lookup(3), selabel\_stats(3), selinux(8)**