

NAME

`selabel_lookup_best_match` – obtain a best match SELinux security context – Only supported on file backend.

SYNOPSIS

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

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

```
int selabel_lookup_best_match(struct selabel_handle *hnd,
                             char **context,
                             const char *key,
                             const char **links,
                             int type);
```

```
int selabel_lookup_best_match_raw(struct selabel_handle *hnd,
                                  char **context,
                                  const char *key,
                                  const char **links,
                                  int type);
```

DESCRIPTION

`selabel_lookup_best_match()` performs a best match lookup operation on the handle *hnd*, returning the result in the memory pointed to by *context*, which must be freed by the caller using `freecon(3)`. The *key* parameter is a file path to check for best match using zero or more *link* (aliases) parameters. The order of precedence for best match is:

1. An exact match for the real path (*key*) or
2. An exact match for any of the *links* (aliases), or
3. The longest fixed prefix match.

The *type* parameter is an optional file *mode* argument that should be set to the mode bits of the file, as determined by `lstat(2)`. *mode* may be zero, however full matching may not occur.

`selabel_lookup_best_match_raw()` behaves identically to `selabel_lookup_best_match()` but does not perform context translation.

RETURN VALUE

On success, zero is returned. On error, `-1` is returned and *errno* is set appropriately.

ERRORS**ENOENT**

No context corresponding to the input *key* and *type* was found.

EINVAL

The *key* and/or *type* inputs are invalid, or the context being returned failed validation.

ENOMEM

An attempt to allocate memory failed.

NOTES

Example usage - When a service creates a device node, it may also create one or more symlinks to the device node. These symlinks may be the only stable name for the device, e.g. if the partition is dynamically assigned. The file label backend supports this by looking up the "best match" for a device node based on its real path (*key*) and any *links* to it (aliases). The order of precedence for best match is described above.

SEE ALSO

selabel_open(3), selabel_stats(3), selinux_set_callback(3), selinux(8), lstat(2), selabel_file(5)