NAME

get_ordered_context_list, get_ordered_context_list_with_level, get_default_context, get_default_context_with_level, get_default_context_with_role, get_default_context_with_rolelevel, query_user_context, manual_user_enter_context, get_default_role − determine SELinux context(s) for user sessions

SYNOPSIS

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

int get_ordered_context_list(const char ****user*, const char ****fr omcon***, char ******list***);**

int get_ordered_context_list_with_level(const char ****user*, const char ****level***, const char ****fr omcon***, char ******list***);**

int get_default_context(const char ****user*, const char ****fr omcon***, char *****newcon***);**

int get_default_context_with_level(const char ****user*, const char ****level***, const char ****fr omcon***, char *****newcon***);**

int get_default_context_with_role(const char ****user*, const char ****role***, const char ****fr omcon***, char *****newcon***);**

int get_default_context_with_rolelevel(const char ****user*, const char ****role***, const char ****level***, const char ****fr omcon***, char *****newcon***);**

int query_user_context(char *****list*, char *****newcon***);**

int manual_user_enter_context(const char ****user*, char *****newcon***);**

int get_default_type(const char ****role*, char *****type***);**

DESCRIPTION

This family of functions can be used to obtain either a prioritized list of all reachable security contexts for a given SELinux user or a single default (highest priority) context for a given SELinux user for use by login-like programs. These functions takes a SELinux user identity that must be defined in the SELinux policy as their input, not a Linux username. Most callers should typically first call **getseuserbyname(3)** to look up the SELinux user identity and level for a given Linux username and then invoke one of **get_ordered_context_list_with_level**() or **get_default_context_with_level**() with the returned SELinux user and level as inputs.

get_ordered_context_list() obtains the list of contexts for the specified SELinux *user* identity that are reachable from the specified *fr omcon* context based on the global */etc/selinux/{SELINUXTYPE}/contexts/default_contexts* file and the per-user */etc/selinux/{SELINUXTYPE}/contexts/users/<username>* file if it exists. The*fr omcon* parameter may be NULL to indicate that the current context should be used. The function returns the number of contexts in the list, or −1 upon errors. The list must be freed using the **freeconary**(3) function.

get_ordered_context_list_with_level() invokes the **get_ordered_context_list**() function and applies the specified level.

get_default_context() is the same as **get_ordered_context_list**() but only returns a single context which has to be freed with **freecon**(3).

get_default_context_with_level() invokes the **get_default_context**() function and applies the specified level.

get_default_context_with_role() is the same as **get_default_context**() but only returns a context with the specified role, returning −1 if no such context is reachable for the user.

get_default_context_with_rolelevel() invokes the **get_default_context_with_role**() function and applies the specified level.

query_user_context() takes a list of contexts, queries the user via stdin/stdout as to which context they want, and returns a new context as selected by the user (which has to be freed with **freecon**(3)).

manual_user_enter_context() allows the user to manually enter a context as a fallback if a list of authorized contexts could not be obtained. Caller must free via **freecon**(3).

get_default_type() Get the default type (domain) for *role* and set *type* to refer to it, which has to be freed with free.

RETURN VALUE

get_ordered_context_list() and **get_ordered_context_list_with_level**() return the number of contexts in the list upon success or −1 upon errors. The other functions return 0 for success or −1 for errors.

SEE ALSO

selinux(8), **freeconary**(3), **freecon**(3), **security_compute_av**(3), **getseuserbyname**(3)