NAME

getexeccon, setexeccon − get or set the SELinux security context used for executing a new process

rpm_execcon − run a helper for rpm in an appropriate security context

SYNOPSIS

#include <selinux/selinux.h>

int getexeccon(char *****context*);**

int getexeccon_raw(char *****context*);**

int setexeccon(char ****context*);**

int setexeccon_raw(char ****context*);**

int setexecfilecon(const char ****filename*, const char ****fallbac k_type***);**

int rpm_execcon(unsigned int *verified***, const char ****filename***, char *const** *ar gv***[] , char *const** *envp***[]);**

DESCRIPTION

getexeccon() retrieves the context used for executing a new process. This returned context should be freed with **freecon**(3) if non-NULL. **getexeccon**() sets ***conte xt** to NULL if no exec context has been explicitly set by the program (i.e. using the default policy behavior).

setexeccon() sets the context used for the next **execve**(2) call. NULL can be passed to **setexeccon**() to reset to the default policy behavior. The exec context is automatically reset after the next **execve**(2), so a program doesn't need to explicitly sanitize it upon startup.

setexeccon() can be applied prior to library functions that internally perform an **execve**(2), e.g. **execl***(3), **execv***(3), **popen**(3), in order to set an exec context for that operation.

getexeccon_raw() and **setexeccon_raw**() behave identically to their non-raw counterparts but do not perform context translation.

Note: Signal handlers that perform an **execve**(2) must take care to save, reset, and restore the exec context to avoid unexpected behavior.

setexecfilecon() sets the context used for the next **execve**(2) call, based on the policy for the *filename*, and falling back to a new context with a *fallbac k_type* in case there is no transition.

rpm_execcon() is deprecated; please use **setexecfilecon**() in conjunction with **execve**(2) in all new code. This function runs a helper for rpm in an appropriate security context. The verified parameter should contain the return code from the signature verification (0 == ok, 1 == notfound, 2 == verifyfail, 3 == nottrusted, 4 == nokey), although this information is not yet used by the function. The function determines the proper security context for the helper based on policy, sets the exec context accordingly, and then executes the specified filename with the provided argument and environment arrays.

RETURN VALUE

On error −1 is returned.

On success **getexeccon**(), **setexeccon**() and **setexecfilecon**() return 0. **r pm_execcon**() only returns upon errors, as it calls **execve**(2).

SEE ALSO

selinux(8), **freecon**(3), **getcon**(3)