NAME

security_getenforce, security_setenforce, security_deny_unknown, security_reject_unknown, security_get_checkreqprot − get or set the enforcing state of SELinux

SYNOPSIS

#include <selinux/selinux.h>

int security_getenforce(void);

int security_setenforce(int *value***);**

int security_deny_unknown(void);

int security_reject_unknown(void);

int security_get_checkreqprot(void);

DESCRIPTION

security_getenforce() returns 0 if SELinux is running in permissive mode, 1 if it is running in enforcing mode, and −1 on error.

security_setenforce() sets SELinux to enforcing mode if the value 1 is passed in, and sets it to permissive mode if 0 is passed in. On success 0 is returned, on error −1 is returned.

security_deny_unknown() returns 0 if SELinux treats policy queries on undefined object classes or permissions as being allowed, 1 if such queries are denied, and −1 on error.

security_reject_unknown() returns 1 if the current policy was built with handle-unknown=reject and SELinux would reject loading it, if it did not define all kernel object classes and permissions. In this state, when **selinux_set_mapping()** and **selinux_check_access()** are used with an undefined userspace class or permission, an error is returned and errno is set to EINVAL.

It returns 0 if the current policy was built with handle-unknown=allow or handle-unknown=deny. In this state, policy queries are treated according to **security_deny_unknown().** −1 is returned on error.

security_get_checkreqprot() can be used to determine whether SELinux is configured to check the protection requested by the application or the actual protection that will be applied by the kernel (including the effects of READ_IMPLIES_EXEC) on mmap and mprotect calls. It returns 0 if SELinux checks the actual protection, 1 if it checks the requested protection, and −1 on error.

SEE ALSO

selinux(8)