NAME

avc_add_callback − additional event notification for SELinux userspace object managers

SYNOPSIS

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

int avc_add_callback(int (*callback)(uint32_t event,
                                     security_id_t ssid,
                                     security_id_t tsid,
                                     security_class_t tclass,
                                     access_vector_t perms,
                                     access_vector_t *out_retained),
                     uint32_t events, security_id_t ssid,
                     security_id_t tsid, security_class_t tclass,
                     access_vector_t perms);
```

DESCRIPTION

avc_add_callback() is used to register callback functions on security events. The purpose of this functionality is to allow userspace object managers to take additional action when a policy change, usually a policy reload, causes permissions to be granted or revoked.

events is the bitwise-*or* of security events on which to register the callback; see **SECURITY EVENTS** below.

ssid, *tsid*, *tclass*, and *perms* specify the source and tar get SID's, target class, and specific permissions that the callback wishes to monitor. The special symbol**SECSID_WILD** may be passed as the *sour ce* or *target* and will cause any SID to match.

callback is the callback function provided by the userspace object manager. The*e vent* argument indicates the security event which occurred; the remaining arguments are interpreted according to the event as described below. The return value of the callback should be zero on success, −1 on error with *errno* set appropriately (but see **RETURN VALUE** below).

SECURITY EVENTS

In all cases below, *ssid* and/or *tsid* may be set to **SECSID_WILD**, indicating that the change applies to all source and/or target SID's. Unless otherwise indicated, the*out_r etained* parameter is unused.

AVC_CALLBACK_GRANT

Previously denied permissions are now granted for *ssid*, *tsid* with respect to *tclass*. *perms* indicates the permissions to grant.

AVC_CALLBACK_TRY_REVOKE

Previously granted permissions are now conditionally revoked for *ssid*, *tsid* with respect to *tclass*. *perms* indicates the permissions to revoke. The callback should set*out_r etained* to the subset of *perms* which are retained as migrated permissions. Note that *out_retained* is ignored if the callback returns −1.

AVC_CALLBACK_REVOKE

Previously granted permissions are now unconditionally revoked for *ssid*, *tsid* with respect to *tclass*. *perms* indicates the permissions to revoke.

AVC_CALLBACK_RESET

Indicates that the cache was flushed. The SID, class, and permission arguments are unused and are set to NULL.

AVC_CALLBACK_AUDITALLOW_ENABLE

The permissions given by *perms* should no w be audited when granted for *ssid*, *tsid* with respect to *tclass*.

AVC_CALLBACK_AUDITALLOW_DISABLE

The permissions given by *perms* should no longer be audited when granted for *ssid*, *tsid* with respect to *tclass*.

AVC_CALLBACK_AUDITDENY_ENABLE

The permissions given by *perms* should now be audited when denied for *ssid*, *tsid* with respect to *tclass*.

AVC_CALLBACK_AUDITDENY_DISABLE

The permissions given by *perms* should no longer be audited when denied for *ssid*, *tsid* with respect to *tclass*.

RETURN VALUE

On success, **avc_add_callback**() returns zero. On error, −1 is returned and *errno* is set appropriately.

A return value of −1 from a callback is interpreted as a failed policy operation. If such a return value is encountered, all remaining callbacks registered on the event are called. In threaded mode, the netlink handler thread may then terminate and cause the userspace AVC to return **EINVAL** on all further permission checks until **avc_destroy**(3) is called. In non-threaded mode, the permission check on which the error occurred will return −1 and the value of *errno* encountered to the caller. In both cases, a log message is produced and the kernel may be notified of the error.

ERRORS

ENOMEM

An attempt to allocate memory failed.

NOTES

If the userspace AVC is running in threaded mode, callbacks registered via **avc_add_callback**() may be executed in the context of the netlink handler thread. This will likely introduce synchronization issues requiring the use of locks. See **avc_init**(3).

Support for dynamic revocation and retained permissions is mostly unimplemented in the SELinux kernel module. The only security event that currently gets exercised is **AVC_CALLBACK_RESET**.

AUTHOR

Eamon Walsh <ewalsh@tycho.nsa.gov>

SEE ALSO

avc_init(3), **avc_has_perm**(3), **avc_context_to_sid**(3), **avc_cache_stats**(3), **security_compute_av**(3) **selinux**(8)