NAME

avc_cache_stats, avc_av_stats, avc_sid_stats − obtain userspace SELinux AVC statistics

SYNOPSIS

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

void avc_av_stats(void);

void avc_sid_stats(void);

void avc_cache_stats(struct avc_cache_stats *stats**);**

DESCRIPTION

The userspace AVC maintains two internal hash tables, one to store security ID's and one to cache access decisions.

avc_av_stats() and **avc_sid_stats**() produce log messages indicating the status of the access decision and SID tables, respectively. The messages contain the number of entries in the table, number of hash buckets and number of buckets used, and maximum number of entries in a single bucket.

avc_cache_stats() populates a structure whose fields reflect cache activity:

```
struct avc_cache_stats {
    unsigned    entry_lookups;
    unsigned    entry_hits;
    unsigned    entry_misses;
    unsigned    entry_discards;
    unsigned    cav_lookups;
    unsigned    cav_hits;
    unsigned    cav_probes;
    unsigned    cav_misses;
};
```

entry_lookups
    Number of queries made.

entry_hits
    Number of times a decision was found in the *aeref* argument.

entry_misses
    Number of times a decision was not found in the *aeref* argument.

entry_discards
    Number of times a decision was not found in the *aeref* argument and the *aeref* argument was non-NULL.

cav_lookups
    Number of cache lookups.

cav_hits
    Number of cache hits.

cav_misses
    Number of cache misses.

cav_probes
    Number of entries examined while searching the cache.

NOTES

When the cache is flushed as a result of a call to **avc_reset**() or a policy change notification, the statistics returned by **avc_cache_stats**() are reset to zero. The SID table, however, is left unchanged.

When a policy change notification is received, a call to **avc_av_stats**() is made before the cache is flushed.

AUTHOR

Eamon Walsh <ewalsh@tycho.nsa.gov>

SEE ALSO

avc_init(3), **avc_has_perm**(3), **avc_context_to_sid**(3), **avc_add_callback**(3), **selinux**(8)