NAME

getauevent, getauevnam, getauevnum, getauevnonam, setauevent, endauevent, getauevent_r, getauevnam_r, getauevnum_r – get audit_user entry

SYNOPSIS

cc [*flag* ...] *file* ... **–lbsm –lsocket –lnsl –lintl** [*library* ...]

#include <sys/param.h>
#include <bsm/libbsm.h>

struct au_event_ent *getauevent(void);

struct au_event_ent *getauevnam(char **name***);**

struct au_event_ent *getauevnum(au_event_t *event_number***);**

au_event_t *getauevnonam(char **event_name***);**

void setauevent(void);

void endauevent(void);

struct au_event_ent *getauevent_r(au_event_ent_t *e, void);

struct au_event_ent *getauevnam_r(au_event_ent_t *e, char **name***);**

struct au_event_ent *getauevnum_r(au_event_ent_t *e, au_event_t *event_number***);**

MT-LEVEL

MT-Safe with exceptions.

The functions **getauevent**(), **getauevnam**(), and **getauevnum**() are not MT-Safe; but, there are equivalent functions: **getauevent_r**(), **getauevnam_r**(), and **getauevnum_r**() - all of which provide the same functionality and a MT-Safe function call interface.

AVAILABILITY

The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See **bsmconv**(1M) for more information.

DESCRIPTION

getauevent( ), **getauevnam( ) , getauevnum( ) , getauevent( ) , getauevnam( ) ,** and **getauevnum( ) ,** each return a pointer to an **audit_event** structure.

getauevent( ) and **getauevent_r( )** enumerate audit_event entries: successive calls to these functions will return either successive audit_event entries or NULL.

getauevnam( ) and **getauevnam_r( )** search for an audit_event entry with a given event name *name.*

getauevnum( ) and **getauevnum_r( )** search for an audit_event entry with a given event number *number*.

getauevnonum( ) searches for an audit_event entry with a given event name *name* and returns the corresponding event number.

setauevent( ) ''rewinds'' to the beginning of the enumeration of audit_event entries. Calls to **getauevnam( )**, **getauevnum( ) , getauevnonum( ) , getauevnam_r( ) ,** or **getauevnum_r( )** may leave the enumeration in an indeterminate state; so, **setauevent( )** should be called before the first **getauevent( )** or **getauevent_r( ) .**

endauevent( ) may be called to indicate that **audit_event** processing is complete; the system may then close any open **audit_event file**, deallocate storage, and so forth.

The three functions **getauevent_r( )**, **getauevnam_r( )**, and **getauevnum_r( )**, each take an argument *e* which is a pointer to an **au_event_ent_t.** This pointer is returned on a successful function call. To assure there is enough space for the information returned, the applications programmer should be sure to allocate **AU_EVENT_NAME_MAX** and **AU_EVENT_DESC_MAX** bytes for the **ae_name** and **ac_desc** elements of the **au_event_ent_t** data structure.

The internal representation of an audit_event entry is an **struct au_event_ent** structure defined in **<bsm/libbsm.h>** with the following members:

```
au_event_t   ae_number;
char         *ae_name;
char         *ae_desc;
au_class_t   ae_class;
```

RETURN VALUES

getauevent( ), **getauevnam( )** and **getauevnum( )** return a pointer to a **struct au_event_ent** if it successfully locates the requested entry; otherwise it returns **NULL**.

getauevnonam( ) returns an event number of type **au_event_t** if it successfully enumerates an entry; otherwise it returns **NULL**, indicating it could not find the requested event name.

FILES

/etc/security/audit_event
Maps audit event numbers to audit event names

/etc/passwd Stores user-id to username mappings

SEE ALSO

bsmconv(1M), **getpwnam**(3C), **getauclassent**(3), **audit_class**(4), **audit_event**(4), **passwd**(4)

NOTES

All information for the functions **getauevent ( )** , **getauevnam ( )** , and **getauevnum ( )** is contained in a static area, so it must be copied if it is to be saved.