NAME

setauuser, **endauuser**, **getauuserent**, **getauuserent_r**, **getauusernam**, **getauusernam_r**, **au_user_mask**, **getfauditflags** - look up information from the audit_user database

LIBRARY

Basic Security Module Library (libbsm, -lbsm)

SYNOPSIS

#include <bsm/libbsm.h>

void
setauuser(*void*);

void
endauuser(*void*);

struct au_user_ent *
getauuserent(*void*);

struct au_user_ent *
getauuserent_r(*struct au_user_ent *u*);

struct au_user_ent *
getauusernam(*const char *name*);

struct au_user_ent *
getauusernam_r(*struct au_user_ent *u*, *const char *name*);

int
au_user_mask(*char *username*, *au_mask_t *mask_p*);

int
getfauditflags(*au_mask_t *usremask*, *au_mask_t *usrdmask*, *au_mask_t *lastmask*);

DESCRIPTION

These interfaces may be used to look up information from the audit_user(5) database, which describes per-user audit configuration. Audit user entries are described by a *au_user_ent*, which stores the user's name in *au_name*, events to always audit in *au_always*, and events never to audit *au_never*.

The **getauuserent**() function returns the next user found in the audit_user(5) database, or the first if the

function has not yet been called. NULL will be returned if no further records are available.

The **getauusernam**() function looks up a user by name. NULL will be returned if no matching class can be found.

The **setauuser**() function resets the iterator through the audit_user(5) database, causing the next call to **getauuserent**() to start again from the beginning of the file.

The **endauuser**() function closes the audit_user(5) database, if open.

The **au_user_mask**() function calculates a new session audit mask to be returned via *mask_p* for the user identified by *username*. If the user audit configuration is not found, the default system audit properties returned by getacflg(3) are used. The resulting mask may be set via a call to setaudit(2) or related variants.

The **getfauditflags**() function generates a new process audit state by combining the audit masks passed as parameters with the system audit masks.

SEE ALSO

setaudit(2), getacflg(3), libbsm(3), audit_user(5)

HISTORY

The OpenBSM implementation was created by McAfee Research, the security division of McAfee Inc., under contract to Apple Computer, Inc., in 2004. It was subsequently adopted by the TrustedBSD Project as the foundation for the OpenBSM distribution.

AUTHORS

This software was created by Robert Watson, Wayne Salamon, and Suresh Krishnaswamy for McAfee Research, the security research division of McAfee, Inc., under contract to Apple Computer, Inc.

The Basic Security Module (BSM) interface to audit records and audit event stream format were defined by Sun Microsystems.

BUGS

These routines cannot currently distinguish between an entry not being found and an error accessing the database. The implementation should be changed to return an error via *errno* when NULL is returned.