NAME

au_close, **au_close_buffer**, **au_close_token**, **au_open**, **au_write** - create and commit audit records

LIBRARY

Basic Security Module Library (libbsm, -lbsm)

SYNOPSIS

#include <bsm/libbsm.h>

int
au_open(*void*);

int
au_write(*int d*, *token_t *tok*);

int
au_close(*int d*, *int keep*, *short event*);

int
au_close_buffer(*int d*, *short event*, *u_char *buffer*, *size_t *buflen*);

int
au_close_token(*token_t *tok*, *u_char *buffer*, *size_t *buflen*);

DESCRIPTION

These interfaces allow applications to allocate audit records, construct a record using a series of tokens, and commit the audit record to the system event log. An extension API is also provided to commit the record to an in-memory buffer rather than the system audit log.

The **au_open**() interface allocates a new audit record descriptor.

The **au_write**() interface adds a token to an allocated audit descriptor. When a token has been successfully added to a record, the caller no longer owns the token memory, and does not need to free it directly via a call to au_free_token(3).

The **au_close**() function is used to commit an audit record to the system audit log, or abandon the record. In either cases, all resources associated with the record will be released. The *keep* argument determines the behavior: a value of AU_TO_WRITE causes the record to be committed; a value of AU_TO_NO_WRITE causes it to be abandoned. When the audit record is committed, a BSM header will be inserted before tokens added to the record, using the event identifier passed via *event*, and a

trailer added to the end. Committing a record to the system audit log requires privilege.

The **au_close_buffer**() function writes the resulting record to an in-memory buffer of size **buflen*; it will write back the filled buffer length into the same variable. The argument *event* is the event identifier to use in the record header.

The **au_close_token**() function generates the BSM stream output for a single token, *tok*, in the passed buffer *buffer*. The initial buffer size and resulting data size are passed via **buflen*. The **au_close_token**() function will free the token before returning.

RETURN VALUES

The function **au_open**() returns a non-negative audit record descriptor number on success, or a negative value on failure, along with error information in *errno*.

The functions **au_write**(), **au_close**(), **au_close_buffer**(), and **au_close_token**() return 0 on success, or a negative value on failure, along with error information in *errno*.

SEE ALSO

audit_submit(3), libbsm(3)

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

Currently, **au_open**() does not reserve kernel resources necessary to commit the record to the trail; on systems supporting **au_close**(), the call will block until resources are available to commit the record. However, this leads to the possibility of an action being permitted without the record being guaranteed to go to disk. Ideally, **au_open**() would reserve resources necessary to commit any submitted record, releasing them on **au_close**().