NAME

mac - introduction to the MAC security API

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/mac.h>

In the kernel configuration file:
options MAC

DESCRIPTION

Mandatory Access Control labels describe confidentiality, integrity, and other security attributes of operating system objects, overriding discretionary access control. Not all system objects support MAC labeling, and MAC policies must be explicitly enabled by the administrator. This API, based on POSIX.1e, includes routines to retrieve, manipulate, set, and convert to and from text the MAC labels on files and processes.

MAC labels consist of a set of (name, value) tuples, representing security attributes from MAC policies. For example, this label contains security labels defined by two policies, mac_biba(4) and mac_mls(4):

```
biba/low,mls/low
```

Further syntax and semantics of MAC labels may be found in maclabel(7).

Applications operate on labels stored in *mac_t*, but can convert between this internal format and a text format for the purposes of presentation to uses or external storage. When querying a label on an object, a *mac_t* must first be prepared using the interfaces described in mac_prepare(3), allowing the application to declare which policies it wishes to interrogate. The application writer can also rely on default label names declared in mac.conf(5).

When finished with a *mac_t*, the application must call mac_free(3) to release its storage.

The following functions are defined:

mac_is_present()
This function, described in mac_is_present(3), allows applications to test whether MAC is configured, as well as whether specific policies are configured.

mac_get_fd(), **mac_get_file**(), **mac_get_link**(), **mac_get_peer**()
These functions, described in mac_get(3), retrieve the MAC labels associated with file descriptors, files, and socket peers.

mac_get_pid(), **mac_get_proc**()
These functions, described in mac_get(3), retrieve the MAC labels associated with processes.

mac_set_fd(), **mac_set_file**(), **mac_set_link**()
These functions, described in mac_set(3), set the MAC labels associated with file descriptors and files.

mac_set_proc()
This function, described in mac_set(3), sets the MAC label associated with the current process.

mac_free()
This function, described in mac_free(3), frees working MAC label storage.

mac_from_text()
This function, described in mac_text(3), converts a text-form MAC label into working MAC label storage, *mac_t*.

mac_prepare(), **mac_prepare_file_label**(), **mac_prepare_ifnet_label**(), **mac_prepare_process_label**(), **mac_prepare_type**()
These functions, described in mac_prepare(3), allocate working storage for MAC label operations. mac_prepare(3) prepares a label based on caller-specified label names; the other calls rely on the default configuration specified in mac.conf(5).

mac_to_text()
This function is described in mac_text(3), and may be used to convert a *mac_t* into a text-form MAC label.

FILES

/etc/mac.conf MAC library configuration file, documented in mac.conf(5). Provides default behavior for applications aware of MAC labels on system objects, but without policy-specific knowledge.

SEE ALSO

mac_free(3), mac_get(3), mac_is_present(3), mac_prepare(3), mac_set(3), mac_text(3), posix1e(3), mac(4), mac.conf(5), mac(9)

STANDARDS

These APIs are loosely based on the APIs described in POSIX.1e, as described in IEEE POSIX.1e draft 17. However, the resemblance of these APIs to the POSIX APIs is loose, as the POSIX APIs were unable to express some notions required for flexible and extensible access control.

HISTORY

Support for Mandatory Access Control was introduced in FreeBSD 5.0 as part of the TrustedBSD Project.