

**NAME**

**mac\_set\_file**, **mac\_set\_fd**, **mac\_set\_proc** - set the MAC label for a file or process

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

```
#include <sys/mac.h>
```

*int*

```
mac_set_file(const char *path, mac_t label);
```

*int*

```
mac_set_link(const char *path, mac_t label);
```

*int*

```
mac_set_fd(int fd, mac_t label);
```

*int*

```
mac_set_proc(mac_t label);
```

**DESCRIPTION**

The **mac\_set\_file()** and **mac\_set\_fd()** functions associate a MAC label specified by *label* to the file referenced to by *path\_p*, or to the file descriptor *fd*, respectively. Note that when a file descriptor references a socket, label operations on the file descriptor act on the socket, not on the file that may have been used as a rendezvous when binding the socket. The **mac\_set\_link()** function is the same as **mac\_set\_file()**, except that it does not follow symlinks.

The **mac\_set\_proc()** function associates the MAC label specified by *label* to the calling process.

A process is allowed to set a label for a file only if it has MAC write access to the file, and its effective user ID is equal to the owner of the file, or has appropriate privileges.

**RETURN VALUES**

The **mac\_set\_fd()**, **mac\_set\_file()**, **mac\_set\_link()**, and **mac\_set\_proc()** functions return the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

**ERRORS**

[EACCES]           MAC write access to the file is denied.

- [EBADF]           The *fd* argument is not a valid file descriptor.
- [EINVAL]           The *label* argument is not a valid MAC label, or the object referenced by *fd* is not appropriate for label operations.
- [EOPNOTSUPP]       Setting MAC labels is not supported by the file referenced by *fd*.
- [EPERM]            The calling process had insufficient privilege to change the MAC label.
- [EROFS]            File system for the object being modified is read only.
- [ENAMETOOLONG]     The length of the pathname in *path\_p* exceeds PATH\_MAX, or a component of the pathname is longer than NAME\_MAX.
- [ENOENT]           The file referenced by *path\_p* does not exist.
- [ENOTDIR]          A component of the pathname referenced by *path\_p* is not a directory.

**SEE ALSO**

mac(3), mac\_free(3), mac\_get(3), mac\_is\_present(3), mac\_prepare(3), mac\_text(3), posix1e(3), mac(4), mac(9)

**HISTORY**

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