NAME

cap_enter, **cap_getmode** — Capability mode system calls

LIBRARY

Standard C Library (libc, −lc)

SYNOPSIS

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

int
cap_enter(void);

int
cap_getmode(u_int *modep);
```

DESCRIPTION

cap_enter() places the current process into capability mode, a mode of execution in which processes may only issue system calls operating on file descriptors or reading limited global system state. Access to global name spaces, such as file system or IPC name spaces, is prevented. If the process is already in a capability mode sandbox, the system call is a no-op. Future process descendants create with `fork`(2) or `pdfork`(2) will be placed in capability mode from inception.

When combined with capabilities created with `cap_new`(2), **cap_enter**() may be used to create kernel-enforced sandboxes in which appropriately-crafted applications or application components may be run.

cap_getmode() returns a flag indicating whether or not the process is in a capability mode sandbox.

CAVEAT

Creating effective process sandboxes is a tricky process that involves identifying the least possible rights required by the process and then passing those rights into the process in a safe manner. See the CAVEAT section of `cap_new`(2) for why this is particularly tricky with UNIX file descriptors as the canonical representation of a right. Consumers of **cap_enter**() should also be aware of other inherited rights, such as access to VM resources, memory contents, and other process properties that should be considered. It is advisable to use `fexecve`(2) to create a runtime environment inside the sandbox that has as few implicitly acquired rights as possible.

RETURN VALUES

The **cap_enter**() and **cap_getmode**() functions return the value 0 if successful; otherwise the value −1 is returned and the global variable *errno* is set to indicate the error.

SEE ALSO

`cap_new`(2), `fexecve`(2), `capsicum`(4)

HISTORY

Support for capabilities and capabilities mode was developed as part of the TrustedBSD Project.

AUTHORS

These functions and the capability facility were created by Robert N. M. Watson at the University of Cambridge Computer Laboratory with support from a grant from Google, Inc.