

NAME

Capsicum - lightweight OS capability and sandbox framework

SYNOPSIS

options **CAPABILITY_MODE**

options **CAPABILITIES**

DESCRIPTION

Capsicum is a lightweight OS capability and sandbox framework implementing a hybrid capability system model. Capabilities are unforgeable tokens of authority that can be delegated and must be presented to perform an action. **Capsicum** makes file descriptors into capabilities.

Capsicum can be used for application and library compartmentalisation, the decomposition of larger bodies of software into isolated (sandboxed) components in order to implement security policies and limit the impact of software vulnerabilities.

Capsicum provides two core kernel primitives:

capability mode

A process mode, entered by invoking `cap_enter(2)`, in which access to global OS namespaces (such as the file system and PID namespaces) is restricted; only explicitly delegated rights, referenced by memory mappings or file descriptors, may be used. Once set, the flag is inherited by future children processes, and may not be cleared.

capabilities

Limit operations that can be called on file descriptors. For example, a file descriptor returned by `open(2)` may be refined using `cap_rights_limit(2)` so that only `read(2)` and `write(2)` can be called, but not `fchmod(2)`. The complete list of the capability rights can be found in the `rights(4)` manual page.

In some cases, **Capsicum** requires use of alternatives to traditional POSIX APIs in order to name objects using capabilities rather than global namespaces:

process descriptors

File descriptors representing processes, allowing parent processes to manage child processes without requiring access to the PID namespace; described in greater detail in `procdesc(4)`.

anonymous shared memory

An extension to the POSIX shared memory API to support anonymous swap objects associated with file descriptors; described in greater detail in `shm_open(2)`.

In some cases, **Capsicum** limits the valid values of some parameters to traditional APIs in order to restrict access to global namespaces:

process IDs

Processes can only act upon their own process ID with syscalls such as `cpuset_setaffinity(2)`.

SEE ALSO

`cap_enter(2)`, `cap_fcntls_limit(2)`, `cap_getmode(2)`, `cap_ioctls_limit(2)`, `cap_rights_limit(2)`, `fchmod(2)`, `open(2)`, `pdfork(2)`, `pdgetpid(2)`, `pdkill(2)`, `pdwait4(2)`, `read(2)`, `shm_open(2)`, `write(2)`, `cap_rights_get(3)`, `libcasper(3)`, `procdesc(4)`

HISTORY

Capsicum first appeared in FreeBSD 9.0, and was developed at the University of Cambridge.

AUTHORS

Capsicum was developed by Robert Watson <rwatson@FreeBSD.org> and Jonathan Anderson <jonathan@FreeBSD.org> at the University of Cambridge, and Ben Laurie <benl@FreeBSD.org> and Kris Kennaway <kris@FreeBSD.org> at Google, Inc., and Pawel Jakub Dawidek <pawel@dawidek.net>.