NAME

Capability rights - Capsicum capability rights for file descriptors

DESCRIPTION

When a file descriptor is created by a function such as accept(2), accept4(2), fhopen(2), kqueue(2), mq_open(2), open(2), openat(2), pdfork(2), pipe(2), shm_open(2), socket(2) or socketpair(2), it is assigned all capability rights. Those rights can be reduced (but never expanded) by using the cap_rights_limit(2), cap_fcntls_limit(2) and cap_ioctls_limit(2) system calls. Once capability rights are reduced, operations on the file descriptor will be limited to those permitted by rights.

The complete list of capability rights is provided below. The *cap_rights_t* type is used to store list of capability rights. The cap_rights_init(3) family of functions should be used to manage the structure.

RIGHTS

The following rights may be specified in a rights mask:

CAP_ACCEPT	Permit accept(2) and accept4(2).
CAP_ACL_CHECK	Permit acl_valid_fd_np(3).
CAP_ACL_DELETE	Permit acl_delete_fd_np(3).
CAP_ACL_GET	Permit acl_get_fd(3) and acl_get_fd_np(3).
CAP_ACL_SET	Permit acl_set_fd(3) and acl_set_fd_np(3).
CAP_BIND	When not in capabilities mode, permit bind(2) and bindat(2) with special value AT_FDCWD in the *fd* parameter. Note that sockets can also become bound implicitly as a result of connect(2) or send(2), and that socket options set with setsockopt(2) may also affect binding behavior.
CAP_BINDAT	Permit bindat(2). This right has to be present on the directory descriptor. This right includes the CAP_LOOKUP right.
CAP_CHFLAGSAT	An alias to CAP_FCHFLAGS and CAP_LOOKUP.
CAP_CONNECT	When not in capabilities mode, permit connect(2) and connectat(2) with special value AT_FDCWD in the *fd* parameter. This right is also required for sendto(2) with a non-NULL destination address.

CAP_CONNECTAT	Permit connectat(2). This right has to be present on the directory descriptor. This right includes the CAP_LOOKUP right.
CAP_CREATE	Permit openat(2) with the O_CREAT flag.
CAP_EVENT	Permit select(2), poll(2), and kevent(2) to be used in monitoring the file descriptor for events.
CAP_EXTATTR_DELETE	Permit extattr_delete_fd(2).
CAP_EXTATTR_GET	Permit extattr_get_fd(2).
CAP_EXTATTR_LIST	Permit extattr_list_fd(2).
CAP_EXTATTR_SET	Permit extattr_set_fd(2).
CAP_FCHDIR	Permit fchdir(2).
CAP_FCHFLAGS	Permit fchflags(2) and chflagsat(2) if the CAP_LOOKUP right is also present.
CAP_FCHMOD	Permit fchmod(2) and fchmodat(2) if the CAP_LOOKUP right is also present.
CAP_FCHMODAT	An alias to CAP_FCHMOD and CAP_LOOKUP.
CAP_FCHOWN	Permit fchown(2) and fchownat(2) if the CAP_LOOKUP right is also present.
CAP_FCHOWNAT	An alias to CAP_FCHOWN and CAP_LOOKUP.
CAP_FCNTL	Permit fcntl(2). Note that only the F_GETFL, F_SETFL, F_GETOWN and F_SETOWN commands require this capability right. Also note that the list of permitted commands can be further limited with the cap_fcntls_limit(2) system call.
CAP_FEXECVE	Permit fexecve(2) and openat(2) with the O_EXEC flag; CAP_READ is also required.
CAP_FLOCK	Permit flock(2), fcntl(2) (with F_GETLK, F_SETLK, F_SETLKW or

	F_SETLK_REMOTE flag) and openat(2) (with O_EXLOCK or O_SHLOCK flag).
CAP_FPATHCONF	Permit fpathconf(2).
CAP_FSCK	Permit UFS background-fsck operations on the descriptor.
CAP_FSTAT	Permit fstat(2) and fstatat(2) if the CAP_LOOKUP right is also present.
CAP_FSTATAT	An alias to CAP_FSTAT and CAP_LOOKUP.
CAP_FSTATFS	Permit fstatfs(2).
CAP_FSYNC	Permit aio_fsync(2), fdatasync(2), fsync(2) and openat(2) with O_FSYNC or O_SYNC flag.
CAP_FTRUNCATE	Permit ftruncate(2) and openat(2) with the O_TRUNC flag.
CAP_FUTIMES	Permit futimens(2) and futimes(2), and permit futimesat(2) and utimensat(2) if the CAP_LOOKUP right is also present.
CAP_FUTIMESAT	An alias to CAP_FUTIMES and CAP_LOOKUP.
CAP_GETPEERNAME	Permit getpeername(2).
CAP_GETSOCKNAME	Permit getsockname(2).
CAP_GETSOCKOPT	Permit getsockopt(2).
CAP_IOCTL	Permit ioctl(2). Be aware that this system call has enormous scope, including potentially global scope for some objects. The list of permitted ioctl commands can be further limited with the cap_ioctls_limit(2) system call.
CAP_KQUEUE	An alias to CAP_KQUEUE_CHANGE and CAP_KQUEUE_EVENT.
CAP_KQUEUE_CHANGE	Permit kevent(2) on a kqueue(2) descriptor that modifies list of monitored events (the *changelist* argument is non-NULL).
CAP_KQUEUE_EVENT	Permit kevent(2) on a kqueue(2) descriptor that monitors events (the

	eventlist argument is non-NULL). CAP_EVENT is also required on file descriptors that will be monitored using kevent(2).
CAP_LINKAT_SOURCE	Permit linkat(2) on the source directory descriptor. This right includes the CAP_LOOKUP right. Warning: CAP_LINKAT_SOURCE makes it possible to link files in a directory for which file descriptors exist that have additional rights. For example, a file stored in a directory that does not allow CAP_READ may be linked in another directory that does allow CAP_READ, thereby granting read access to a file that is otherwise unreadable.
CAP_LINKAT_TARGET	Permit linkat(2) on the target directory descriptor. This right includes the CAP_LOOKUP right.
CAP_LISTEN	Permit listen(2); not much use (generally) without CAP_BIND.
CAP_LOOKUP	Permit the file descriptor to be used as a starting directory for calls such as linkat(2), openat(2), and unlinkat(2).
CAP_MAC_GET	Permit mac_get_fd(3).
CAP_MAC_SET	Permit mac_set_fd(3).
CAP_MKDIRAT	Permit mkdirat(2). This right includes the CAP_LOOKUP right.
CAP_MKFIFOAT	Permit mkfifoat(2). This right includes the CAP_LOOKUP right.
CAP_MKNODAT	Permit mknodat(2). This right includes the CAP_LOOKUP right.
CAP_MMAP	Permit mmap(2) with the PROT_NONE protection.
CAP_MMAP_R	Permit mmap(2) with the PROT_READ protection. This right includes the CAP_READ and CAP_SEEK rights.
CAP_MMAP_RW	An alias to CAP_MMAP_R and CAP_MMAP_W.
CAP_MMAP_RWX	An alias to CAP_MMAP_R, CAP_MMAP_W and CAP_MMAP_X.
CAP_MMAP_RX	An alias to CAP_MMAP_R and CAP_MMAP_X.

CAP_MMAP_W	Permit mmap(2) with the PROT_WRITE protection. This right includes the CAP_WRITE and CAP_SEEK rights.
CAP_MMAP_WX	An alias to CAP_MMAP_W and CAP_MMAP_X.
CAP_MMAP_X	Permit mmap(2) with the PROT_EXEC protection. This right includes the CAP_SEEK right.
CAP_PDGETPID	Permit pdgetpid(2).
CAP_PDKILL	Permit pdkill(2).
CAP_PEELOFF	Permit sctp_peeloff(2).
CAP_PREAD	An alias to CAP_READ and CAP_SEEK.
CAP_PWRITE	An alias to CAP_SEEK and CAP_WRITE.
CAP_READ	Permit aio_read(2) (CAP_SEEK is also required), openat(2) with the O_RDONLY flag, read(2), readv(2), recv(2), recvfrom(2), recvmsg(2), pread(2) (CAP_SEEK is also required), preadv(2) (CAP_SEEK is also required) and related system calls.
CAP_RECV	An alias to CAP_READ.
CAP_RENAMEAT_SOURCE	Permit renameat(2) on the source directory descriptor. This right includes the CAP_LOOKUP right. Warning: CAP_RENAMEAT_SOURCE makes it possible to move files to a directory for which file descriptors exist that have additional rights. For example, a file stored in a directory that does not allow CAP_READ may be moved to another directory that does allow CAP_READ, thereby granting read access to a file that is otherwise unreadable.
CAP_RENAMEAT_TARGET	Permit renameat(2) on the target directory descriptor. This right includes the CAP_LOOKUP right.
CAP_SEEK	Permit operations that seek on the file descriptor, such as lseek(2), but also required for I/O system calls that can read or write at any position in the file, such as pread(2) and pwrite(2).

CAP_SEM_GETVALUE	Permit sem_getvalue(3).
CAP_SEM_POST	Permit sem_post(3).
CAP_SEM_WAIT	Permit sem_wait(3) and sem_trywait(3).
CAP_SEND	An alias to CAP_WRITE.
CAP_SETSOCKOPT	Permit setsockopt(2); this controls various aspects of socket behavior and may affect binding, connecting, and other behaviors with global scope.
CAP_SHUTDOWN	Permit explicit shutdown(2); closing the socket will also generally shut down any connections on it.
CAP_SYMLINKAT	Permit symlinkat(2). This right includes the CAP_LOOKUP right.
CAP_TTYHOOK	Allow configuration of TTY hooks, such as snp(4), on the file descriptor.
CAP_UNLINKAT	Permit unlinkat(2) and renameat(2). This right is only required for renameat(2) on the destination directory descriptor if the destination object already exists and will be removed by the rename. This right includes the CAP_LOOKUP right.
CAP_WRITE	Allow aio_write(2), openat(2) with O_WRONLY and O_APPEND flags set, send(2), sendmsg(2), sendto(2), write(2), writev(2), pwrite(2), pwritev(2) and related system calls. For sendto(2) with a non-NULL connection address, CAP_CONNECT is also required. For openat(2) with the O_WRONLY flag, but without the O_APPEND flag, CAP_SEEK is also required. For aio_write(2), pwrite(2) and pwritev(2) CAP_SEEK is also required.

SEE ALSO

accept(2), accept4(2), aio_fsync(2), aio_read(2), aio_write(2), bind(2), bindat(2), cap_enter(2), cap_fcntls_limit(2), cap_ioctls_limit(2), cap_rights_limit(2), chflagsat(2), connect(2), connectat(2), extattr_delete_fd(2), extattr_get_fd(2), extattr_list_fd(2), extattr_set_fd(2), fchflags(2), fchmod(2), fchmodat(2), fchown(2), fchownat(2), fcntl(2), fexecve(2), fhopen(2), flock(2), fpathconf(2), fstat(2), fstatat(2), fstatfs(2), fsync(2), ftruncate(2), futimes(2), getpeername(2), getsockname(2), getsockopt(2), ioctl(2), kevent(2), kqueue(2), linkat(2), listen(2), mmap(2), mq_open(2), open(2), openat(2), pdfork(2),

pdgetpid(2), pdkill(2), pdwait4(2), pipe(2), poll(2), pread(2), preadv(2), pwrite(2), pwritev(2), read(2), readv(2), recv(2), recvfrom(2), recvmsg(2), renameat(2), sctp_peeloff(2), select(2), send(2), sendmsg(2), sendto(2), setsockopt(2), shm_open(2), shutdown(2), socket(2), socketpair(2), symlinkat(2), unlinkat(2), write(2), writev(2), acl_delete_fd_np(3), acl_get_fd(3), acl_get_fd_np(3), acl_set_fd(3), acl_set_fd_np(3), acl_valid_fd_np(3), mac_get_fd(3), mac_set_fd(3), sem_getvalue(3), sem_post(3), sem_trywait(3), sem_wait(3), capsicum(4), snp(4)

HISTORY

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

AUTHORS

This manual page was created by Pawel Jakub Dawidek <*pawel@dawidek.net*> under sponsorship from the FreeBSD Foundation based on the cap_new(2) manual page by Robert Watson <*rwatson@FreeBSD.org*>.