(/) Wiki

Hardened/Overview of POSIX capabilities

From Gentoo Wiki
< Hardened (/wiki/Hardened)
Jump to:navigation Jump to:search
POSIX capabilities are a partitioning of the all powerful root privilege into a set of distinct privileges.

Contents

CAP_CHOWN

CODE **CAP_CHOWN**

```
CAP_CHOWN
        In a system with the [_POSIX_CHOWN_RESTRICTED] option defined,
        this overrides the restriction of changing file ownership and
        group ownership.
```

CAP_DAC_OVERRIDE

CODE **CAP_DAC_OVERRIDE**

```
CAP_DAC_OVERRIDE
        Override all DAC access, including ACL execute access
        if [_POSIX_ACL] is defined.
        Excluding DAC access covered by CAP_LINUX_IMMUTABLE.
```

CAP_DAC_READ_SEARCH

CODE **CAP_DAC_READ_SEARCH**

```
CAP_DAC_READ_SEARCH
        Overrides all DAC restrictions, regarding read and search on files
        and directories, including ACL restrictions, if [_POSIX_ACL] is
        defined. Excluding DAC access covered by CAP_LINUX_IMMUTABLE.
```

CAP_FOWNER

CODE **CAP_FOWNER**

```
CAP_FOWNER
        Overrides all restrictions about allowed operations on files, where
        file owner ID must be equal to the user ID, except where CAP_FSETID
        is applicable. It doesn't override MAC and DAC restrictions.
```

CAP_FSETID

CODE **CAP_FSETID**

```
CAP_FSETID
        Overrides the following restrictions, that the effective user ID shall
        match the file owner ID, when setting the S_ISUID and S_ISGID bits on
        that file; that the effective group ID (or one of the supplementary
        group IDs) shall match the file owner ID when setting the S_ISGID bit
        on that file; that the S_ISUID and S_ISGID bits are cleared on
        successful return from chown(2) (not implemented).
```

CAP_FS_MASK

CODE **CAP_FS_MASK**

```
CAP_FS_MASK
        Used to decide between falling back on the old suser() or fsuser().
```

CAP_KILL

CODE **CAP_KILL**

```
CAP_KILL
        Overrides the restriction, that the real or effective user ID of a process,
        sending a signal, must match the real or effective user ID of the process,
        receiving the signal.
```

CAP_SETGID

CODE **CAP_SETGID**

```
CAP_SETGID
        Allows setgid(2) manipulation;
        Allows setgroups(2);
        Allows forged gids on socket credentials passing.
```

CAP_SETUID

CODE **CAP_SETUID**

```
CAP_SETUID
        Allows set*uid(2) manipulation (including fsuid);
        Allows forged pids on socket credentials passing.
```

CAP_SETPCAP

CODE **CAP_SETPCAP**

```
CAP_SETPCAP
        Transfer any capability in your permitted set to any pid, remove any capabili
ty in your permitted set from any pid.
```

CAP_LINUX_IMMUTABLE

CODE **CAP_LINUX_IMMUTABLE**

```
CAP_LINUX_IMMUTABLE
        Allow modification of S_IMMUTABLE and S_APPEND file attributes.
```

CAP_NET_BIND_SERVICE

CODE **CAP_NET_BIND_SERVICE**

```
CAP_NET_BIND_SERVICE
        Allows binding to TCP/UDP sockets below 1024;
        Allows binding to ATM VCIs below 32.
```

CAP_NET_BROADCAST

CODE **CAP_NET_BROADCAST**

```
CAP_NET_BROADCAST
        Allow broadcasting, listen to multicast.
```

CAP_NET_ADMIN

CODE **CAP_NET_ADMIN**

```
CAP_NET_ADMIN
        Allow interface configuration;
        Allow administration of IP firewall, masquerading and accounting;
        Allow setting debug option on sockets;
        Allow modification of routing tables;
        Allow setting arbitrary process / process group ownership on sockets;
        Allow binding to any address for transparent proxying;
        Allow setting TOS (type of service);
        Allow setting promiscuous mode;
        Allow clearing driver statistics;
        Allow multicasting;
        Allow read/write of devicespecific registers;
        Allow activation of ATM control sockets.
```

CAP_NET_RAW

CODE **CAP_NET_RAW**

```
CAP_NET_RAW
        Allow use of RAW sockets;
        Allow use of PACKET sockets.
```

CAP_IPC_LOCK

CODE **CAP_IPC_LOCK**

```
CAP_IPC_LOCK
        Allow locking of shared memory segments;
        Allow mlock and mlockall (which doesn't really have anything to do with IPC).
```

CAP_IPC_OWNER

CODE **CAP_IPC_OWNER**

```
CAP_IPC_OWNER
        Override IPC ownership checks.
```

CAP_SYS_MODULE

CODE **CAP_SYS_MODULE**

```
CAP_SYS_MODULE
        Insert and remove kernel modules  modify kernel without limit;
        Modify cap_bset.
```

CAP_SYS_RAWIO

CODE **CAP_SYS_RAWIO**

```
CAP_SYS_RAWIO
        Allow ioperm/iopl access;
        Allow sending USB messages to any device via /proc/bus/usb.
```

CAP_SYS_CHROOT

CODE **CAP_SYS_CHROOT**

```
CAP_SYS_CHROOT
        Allow use of chroot().
```

CAP_SYS_PTRACE

CODE **CAP_SYS_PTRACE**

```
CAP_SYS_PTRACE
        Allow ptrace() of any process.
```

CAP_SYS_PACCT

CODE **CAP_SYS_PACCT**

```
CAP_SYS_PACCT
        Allow configuration of process accounting.
```

CAP_SYS_ADMIN

CODE **CAP_SYS_ADMIN**

```
CAP_SYS_ADMIN
        Allow configuration of the secure attention key;
        Allow administration of the random device;
        Allow examination and configuration of disk quotas;
        Allow configuring the kernel's syslog (printk behaviour);
        Allow setting the domainname;
        Allow setting the hostname;
        Allow calling bdflush();
        Allow mount() and umount(), setting up new smb connection;
        Allow some autofs root ioctls;
        Allow nfsservctl; Allow VM86_REQUEST_IRQ;
        Allow to read/write pci config on alpha; Allow irix_prctl on mips (setstacksi
ze);
        Allow flushing all cache on m68k (sys_cacheflush);
        Allow removing semaphores; Used instead of CAP_CHOWN to "chown" IPC message q
ueues, semaphores and shared memory;
        Allow locking/unlocking of shared memory segment;
        Allow turning swap on/off;
        Allow forged pids on socket credentials passing;
        Allow setting readahead and flushing buffers on block devices;
        Allow setting geometry in floppy driver;
        Allow turning DMA on/off in xd driver;
        Allow administration of md devices (mostly the above, but some extra ioctls);
        Allow tuning the ide driver;
        Allow access to the nvram device;
        Allow administration of apm_bios, serial and bttv (TV) device;
        Allow manufacturer commands in isdn CAPI support driver;
        Allow reading nonstandardized portions of pci configuration space;
        Allow DDI debug ioctl on sbpcd driver;
        Allow setting up serial ports;
        Allow sending raw qic117 commands;
        Allow enabling/disabling tagged queuing on SCSI controllers and sending arbit
rary SCSI commands;
        Allow setting encryption key on loopback filesystem.
```

CAP_SYS_BOOT

CODE **CAP_SYS_BOOT**

```
CAP_SYS_BOOT
        Allow use of reboot().
```

CAP_SYS_NICE

CODE **CAP_SYS_NICE**

```
CAP_SYS_NICE
        Allow raising priority and setting priority on other (different UID) processe
s;
        Allow use of FIFO and roundrobin (realtime) scheduling on own processes and s
etting
        the scheduling algorithm used by another process.
```

CAP_SYS_RESOURCE

CODE **CAP_SYS_RESOURCE**

```
CAP_SYS_RESOURCE
        Override resource limits. Set resource limits;
        Override quota limits;
        Override reserved space on ext2 filesystem;
        Modify data journaling mode on ext3 filesystem
        (uses journaling resources); NOTE: ext2 honors fsuid when checking for
        resource overrides, so you can override using fsuid too;
        Override size restrictions on IPC message queues;
        Allow more than 64hz interrupts from the realtime clock;
        Override max number of consoles on console allocation;
        Override max number of keymaps.
```

CAP_SYS_TIME

CODE **CAP_SYS_TIME**

```
CAP_SYS_TIME
        Allow manipulation of system clock;
        Allow irix_stime on mips;
        Allow setting the realtime clock.
```

CAP_SYS_TTY_CONFIG

CODE **CAP_SYS_TTY_CONFIG**

```
CAP_SYS_TTY_CONFIG
        Allow configuration of tty devices; Allow vhangup() of tty.
```

CAP_MKNOD

CODE **CAP_MKNOD**

```
CAP_MKNOD
        Allow the privileged aspects of mknod().
```

CAP_LEASE

CODE **CAP_LEASE**

```
CAP_LEASE
        Allow taking of leases on files.
```

This page is based on a document formerly found on our main website gentoo.org (https://www.gentoo.org/).
The following people contributed to the original document: **Ned Ludd, Adam Mondl**
They are listed here because wiki history does not allow for any external attribution. If you edit the wiki article, please do **not** add yourself here; your contributions are recorded on each article's associated history page.

Retrieved from "https://wiki.gentoo.org/index.php?title=Hardened/Overview_of_POSIX_capabilities&oldid=330422 (https://wiki.gentoo.org/index.php?title=Hardened/Overview_of_POSIX_capabilities&oldid=330422)"