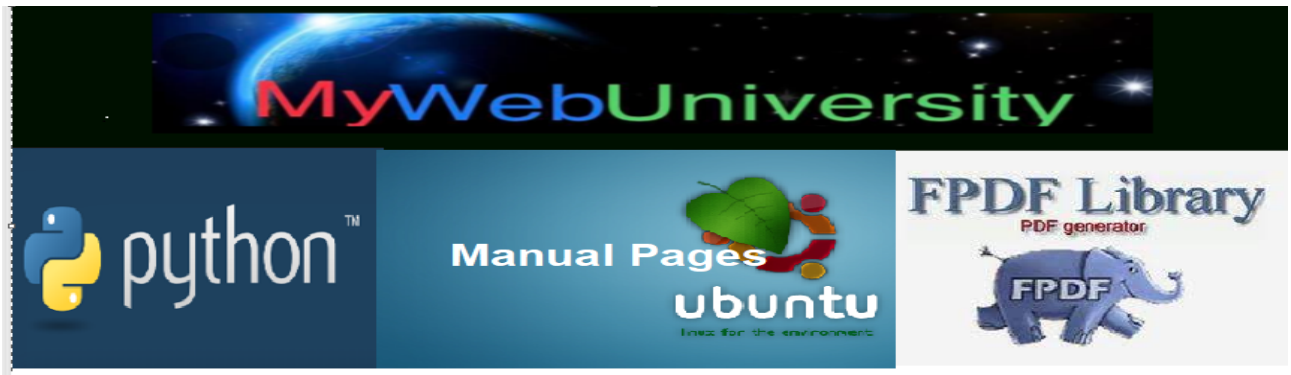

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'landlock_restrict_self.2'

$ man landlock_restrict_self.2

landlock_restrict_self(2) System Calls Manual landlock_restrict_self(2)

NAME

landlock_restrict_self - enforce a Landlock ruleset

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#include <linux/landlock.h>  /* Definition of LANDLOCK_* constants */
#include <sys/syscall.h>     /* Definition of SYS_* constants */
int syscall(SYS_landlock_restrict_self, int ruleset_fd,
            uint32_t flags);
```

DESCRIPTION

Once a Landlock ruleset is populated with the desired rules, the landlock_restrict_self() system call enforces this ruleset on the calling thread. See landlock(7) for a global overview.

A thread can be restricted with multiple rulesets that are then composed together to form the thread's Landlock domain. This can be seen as a stack of rulesets but it is implemented in a more efficient way. A domain can only be updated in such a way that the constraints of each past and future composed rulesets will restrict the thread and its future children for their entire life. It is then possible to gradually enforce tailored access control poli? cies with multiple independent rulesets coming from different sources (e.g., init system configuration, user session policy, built-in application policy). However, most applica? tions should only need one call to landlock_restrict_self() and they should avoid arbitrary numbers of such calls because of the composed rulesets limit. Instead, developers are en?

couraged to build a single tailored ruleset with multiple calls to landlock_add_rule(2).

In order to enforce a ruleset, either the caller must have the CAP_SYS_ADMIN capability in its user namespace, or the thread must already have the no_new_privs bit set. As for sec? comp(2), this avoids scenarios where unprivileged processes can affect the behavior of priv? ileged children (e.g., because of set-user-ID binaries). If that bit was not already set by an ancestor of this thread, the thread must make the following call:

```
prctl(PR_SET_NO_NEW_PRIVS, 1, 0, 0, 0);
```

ruleset_fd is a Landlock ruleset file descriptor obtained with landlock_create_ruleset(2) and fully populated with a set of calls to landlock_add_rule(2).

flags must be 0.

RETURN VALUE

On success, landlock_restrict_self() returns 0. On error, -1 is returned and errno is set to indicate the error.

ERRORS

landlock_restrict_self() can fail for the following reasons:

EOPNOTSUPP
: Landlock is supported by the kernel but disabled at boot time.

EINVAL flags is not 0.

EBADF ruleset_fd is not a file descriptor for the current thread.

EBADFD ruleset_fd is not a ruleset file descriptor.

EPERM ruleset_fd has no read access to the underlying ruleset, or the calling thread is not running with no_new_privs, or it doesn't have the CAP_SYS_ADMIN in its user name? space.

E2BIG The maximum number of composed rulesets is reached for the calling thread. This limit is currently 64.

STANDARDS

Linux.

HISTORY

Linux 5.13.

EXAMPLES

See landlock(7).

SEE ALSO

landlock_create_ruleset(2), landlock_add_rule(2), landlock(7)