NAME

bwrap − container setup utility

SYNOPSIS

bwrap [*OPTION...*] [*COMMAND*]

DESCRIPTION

bwrap is a privileged helper for container setup. You are unlikely to use it directly from the commandline, although that is possible.

It works by creating a new, completely empty, filesystem namespace where the root is on a tmpfs that is invisible from the host, and which will be automatically cleaned up when the last process exits. You can then use commandline options to construct the root filesystem and process environment for the command to run in the namespace.

By default, **bwrap** creates a new mount namespace for the sandbox. Optionally it also sets up new user, ipc, pid, network and uts namespaces (but note the user namespace is required if bwrap is not installed setuid root). The application in the sandbox can be made to run with a different UID and GID.

If needed (e.g. when using a PID namespace) **bwrap** is running a minimal pid 1 process in the sandbox that is responsible for reaping zombies. It also detects when the initial application process (pid 2) dies and reports its exit status back to the original spawner. The pid 1 process exits to clean up the sandbox when there are no other processes in the sandbox left.

OPTIONS

When options are used multiple times, the last option wins, unless otherwise specified.

General options:

−−help
Print help and exit

−−version
Print version

−−args FD
Parse nul−separated arguments from the given file descriptor. This option can be used multiple times to parse options from multiple sources.

Options related to kernel namespaces:

−−unshare−user
Create a new user namespace

−−unshare−user−try
Create a new user namespace if possible else skip it

−−unshare−ipc
Create a new ipc namespace

−−unshare−pid
Create a new pid namespace

−−unshare−net
Create a new network namespace

−−unshare−uts
Create a new uts namespace

−−unshare−cgroup
Create a new cgroup namespace

−−unshare−cgroup−try
Create a new cgroup namespace if possible else skip it

−−unshare−all

Unshare all possible namespaces. Currently equivalent with: **--unshare-user-try --unshare-ipc --unshare-pid --unshare-net --unshare-uts --unshare-cgroup-try**

--uid UID
Use a custom user id in the sandbox (requires **--unshare-user**)

--gid GID
Use a custom group id in the sandbox (requires **--unshare-user**)

--hostname HOSTNAME
Use a custom hostname in the sandbox (requires **--unshare-uts**)

Options about environment setup:

--chdir DIR
Change directory to DIR

--setenv VAR VALUE
Set an environment variable

--unsetenv VAR
Unset an environment variable

Options for monitoring the sandbox from the outside:

--lock-file DEST
Take a lock on DEST while the sandbox is running. This option can be used multiple times to take locks on multiple files.

--sync-fd FD
Keep this file descriptor open while the sandbox is running

Filesystem related options. These are all operations that modify the filesystem directly, or mounts stuff in the filesystem. These are applied in the order they are given as arguments. Any missing parent directories that are required to create a specified destination are automatically created as needed.

--bind SRC DEST
Bind mount the host path SRC on DEST

--bind-try SRC DEST
Equal to **--bind** but ignores non-existent SRC

--dev-bind SRC DEST
Bind mount the host path SRC on DEST, allowing device access

--dev-bind-try SRC DEST
Equal to **--dev-bind** but ignores non-existent SRC

--ro-bind SRC DEST
Bind mount the host path SRC readonly on DEST

--ro-bind-try SRC DEST
Equal to **--ro-bind** but ignores non-existent SRC

--remount-ro DEST
Remount the path DEST as readonly. It works only on the specified mount point, without changing any other mount point under the specified path

--proc DEST
Mount procfs on DEST

--dev DEST
Mount new devtmpfs on DEST

--tmpfs DEST
Mount new tmpfs on DEST

--mqueue DEST

Mount new mqueue on DEST

−−dir DEST
Create a directory at DEST

−−file FD DEST
Copy from the file descriptor FD to DEST

−−bind−data FD DEST
Copy from the file descriptor FD to a file which is bind−mounted on DEST

−−ro−bind−data FD DEST
Copy from the file descriptor FD to a file which is bind−mounted readonly on DEST

−−symlink SRC DEST
Create a symlink at DEST with target SRC

Lockdown options:

−−seccomp FD
Load and use seccomp rules from FD. The rules need to be in the form of a compiled eBPF program, as generated by seccomp_export_bpf.

−−exec−label LABEL
Exec Label from the sandbox. On an SELinux system you can specify the SELinux context for the sandbox process(s).

−−file−label LABEL
File label for temporary sandbox content. On an SELinux system you can specify the SELinux context for the sandbox content.

−−block−fd FD
Block the sandbox on reading from FD until some data is available.

−−userns−block−fd FD
Do not initialize the user namespace but wait on FD until it is ready. This allow external processes (like newuidmap/newgidmap) to setup the user namespace before it is used by the sandbox process.

−−info−fd FD
Write information in JSON format about the sandbox to FD.

−−new−session
Create a new terminal session for the sandbox (calls setsid()). This disconnects the sandbox from the controlling terminal which means the sandbox can't for instance inject input into the terminal.

Note: In a general sandbox, if you don't use −−new−session, it is recommended to use seccomp to disallow the TIOCSTI ioctl, otherwise the application can feed keyboard input to the terminal.

−−die−with−parent
Ensures child process (COMMAND) dies when bwrap's parent dies. Kills (SIGKILL) all bwrap sandbox processes in sequence from parent to child including COMMAND process when bwrap or bwrap's parent dies. See prctl, PR_SET_PDEATHSIG.

−−as−pid−1
Do not create a process with PID=1 in the sandbox to reap child processes.

−−cap−add CAP
Add the specified capability when running as privileged user. It accepts the special value ALL to add all the permitted caps.

−−cap−drop CAP
Drop the specified capability when running as privileged user. It accepts the special value ALL to drop all the caps. By default no caps are left in the sandboxed process. The **−−cap−add** and **−−cap−drop** options are processed in the order they are specified on the command line. Please be careful to the order they are specified.

ENVIRONMENT

HOME

Used as the cwd in the sandbox if **--chdir** has not been explicitly specified and the current cwd is not present inside the sandbox. The **--setenv** option can be used to override the value that is used here.

EXIT STATUS

The **bwrap** command returns the exit status of the initial application process (pid 2 in the sandbox).