NAME

setuid, **seteuid**, **setgid**, **setegid** - set user and group ID

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <unistd.h>

int
setuid(*uid_t uid*);

int
seteuid(*uid_t euid*);

int
setgid(*gid_t gid*);

int
setegid(*gid_t egid*);

DESCRIPTION

The **setuid**() system call sets the real and effective user IDs and the saved set-user-ID of the current process to the specified value. The **setuid**() system call is permitted if the specified ID is equal to the real user ID or the effective user ID of the process, or if the effective user ID is that of the super user.

The **setgid**() system call sets the real and effective group IDs and the saved set-group-ID of the current process to the specified value. The **setgid**() system call is permitted if the specified ID is equal to the real group ID or the effective group ID of the process, or if the effective user ID is that of the super user.

The **seteuid**() system call (**setegid**()) sets the effective user ID (group ID) of the current process. The effective user ID may be set to the value of the real user ID or the saved set-user-ID (see intro(2) and execve(2)); in this way, the effective user ID of a set-user-ID executable may be toggled by switching to the real user ID, then re-enabled by reverting to the set-user-ID value. Similarly, the effective group ID may be set to the value of the real group ID or the saved set-group-ID.

RETURN VALUES

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The system calls will fail if:

[EPERM] The user is not the super user and the ID specified is not the real, effective ID, or saved ID.

SEE ALSO

getgid(2), getuid(2), issetugid(2), setregid(2), setreuid(2)

STANDARDS

The **setuid**() and **setgid**() system calls are compliant with the IEEE Std 1003.1-1990 ("POSIX.1") specification with _POSIX_SAVED_IDS not defined with the permitted extensions from Appendix B.4.2.2. The **seteuid**() and **setegid**() system calls are extensions based on the POSIX concept of _POSIX_SAVED_IDS, and have been proposed for a future revision of the standard.

HISTORY

The **setuid**() function appeared in Version 1 AT&T UNIX. The **setgid**() function appeared in Version 4 AT&T UNIX.

SECURITY CONSIDERATIONS

Read and write permissions to files are determined upon a call to open(2). Once a file descriptor is open, dropping privilege does not affect the process’s read/write permissions, even if the user ID specified has no read or write permissions to the file. These files normally remain open in any new process executed, resulting in a user being able to read or modify potentially sensitive data.

To prevent these files from remaining open after an exec(3) call, be sure to set the close-on-exec flag:

```
void
pseudocode(void)
{
        int fd;
        /* ... */

        fd = open("/path/to/sensitive/data", O_RDWR | O_CLOEXEC);
        if (fd == -1)
                err(1, "open");

        /* ... */
        execve(path, argv, environ);
}
```