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

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'setreuid.3p' command

$ man setreuid.3p

SETREUID(3P) POSIX Programmer's Manual SETREUID(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

setreuid ? set real and effective user IDs

SYNOPSIS

#include <unistd.h>

int setreuid(uid_t ruid, uid_t euid);

DESCRIPTION

The setreuid() function shall set the real and effective user IDs of the current process to the values specified by the ruid and euid argu? ments. If ruid or euid is -1, the corresponding effective or real user ID of the current process shall be left unchanged.

A process with appropriate privileges can set either ID to any value.

An unprivileged process can only set the effective user ID if the euid argument is equal to either the real, effective, or saved user ID of the process.

If the real user ID is being set (ruid is not -1), or the effective user ID is being set to a value not equal to the real user ID, then the saved set-user-ID of the current process shall be set equal to the new

effective user ID.

It is unspecified whether a process without appropriate privileges is permitted to change the real user ID to match the current effective user ID or saved set-user-ID of the process.

RETURN VALUE

Upon successful completion, 0 shall be returned. Otherwise, -1 shall be returned and errno set to indicate the error.

ERRORS

The setreuid() function shall fail if:

EINVAL The value of the ruid or euid argument is invalid or out-of-range.

EPERM The current process does not have appropriate privileges, and either an attempt was made to change the effective user ID to a value other than the real user ID or the saved set-user-ID or an attempt was made to change the real user ID to a value not per? mitted by the implementation.

The following sections are informative.

EXAMPLES

Setting the Effective User ID to the Real User ID

The following example sets the effective user ID of the calling process to the real user ID, so that files created later will be owned by the current user. It also sets the saved set-user-ID to the real user ID, so any future attempt to set the effective user ID back to its previous value will fail.

```
#include <unistd.h>
#include <sys/types.h>
...
setreuid(getuid(), getuid());
...
```

APPLICATION USAGE

None.

RATIONALE

Earlier versions of this standard did not specify whether the saved

set-user-ID was affected by setreuid() calls. This version specifies common existing practice that constitutes an important security fea? ture. The ability to set both the effective user ID and saved set-user-ID to be the same as the real user ID means that any security weakness in code that is executed after that point cannot result in malicious code being executed with the previous effective user ID. Privileged ap? plications could already do this using just setuid(), but for non-priv? ileged applications the only standard method available is to use this feature of setreuid().

FUTURE DIRECTIONS

None.

SEE ALSO

getegid(), geteuid(), getgid(), getuid(), setegid(), seteuid(), set? gid(), setregid(), setuid()

The Base Definitions volume of POSIX.1?2017, <unistd.h>

COPYRIGHT