**Special File Permissions (setuid, setgid and Sticky Bit)**

Three special types of permissions are available for executable files and public directories. When these permissions are set, any user who runs that executable file assumes the user ID of the owner (or group) of the executable file.

You must be extremely careful when you set special permissions, because special permissions constitute a security risk. For example, a user can gain superuser privileges by executing a program that sets the user ID (UID) to root. Also, all users can set special permissions for files they own, which constitutes another security concern.

You should monitor your system for any unauthorized use of the setuid and setgid permissions to gain superuser privileges. To search for and list all of the files that use these permissions, see [How to Find Files With setuid Permissions](https://docs.oracle.com/cd/E19683-01/816-4883/6mb2joatb/index.html). A suspicious listing grants ownership of such a program to a user rather than to root or bin.

**setuid Permission**

When set-user identification (setuid) permission is set on an executable file, a process that runs this file is granted access based on the owner of the file (usually root), rather than the user who is running the executable file. This special permission allows a user to access files and directories that are normally only available to the owner. For example, the setuid permission on the passwd command makes it possible for a user to change passwords, assuming the permissions of the root ID:

|  |
| --- |
| -r-**s**r-sr-x 3 root sys 104580 Sep 16 12:02 /usr/bin/passwd |

This special permission presents a security risk, because some determined users can find a way to maintain the permissions that are granted to them by the setuid process even after the process has finished executing.

**Note –**

The use of setuid permissions with the reserved UIDs (0–100) from a program might not set the effective UID correctly. Use a shell script instead or avoid using the reserved UIDs with setuid permissions.

**setgid Permission**

The set-group identification (setgid) permission is similar to setuid, except that the process's effective group ID (GID) is changed to the group owner of the file, and a user is granted access based on permissions granted to that group. The /usr/bin/mail command has setgid permissions:

|  |
| --- |
| -r-x--**s**--x 1 root mail 63628 Sep 16 12:01 /usr/bin/mail |

When setgid permission is applied to a directory, files that were created in this directory belong to the group to which the directory belongs, not the group to which the creating process belongs. Any user who has write and execute permissions in the directory can create a file there. However, the file belongs to the group that owns the directory, not to the user's group ownership.

You should monitor your system for any unauthorized use of the setuid and setgid permissions to gain superuser privileges. To search for and list all of the files that use these permissions, see [How to Find Files With setuid Permissions](https://docs.oracle.com/cd/E19683-01/816-4883/6mb2joatb/index.html). A suspicious listing grants group ownership of such a program to a user rather than to root or bin.

**Sticky Bit**

The **sticky** **bit** is a permission bit that protects the files within a directory. If the directory has the sticky bit set, a file can be deleted only by the owner of the file, the owner of the directory, or by root. This special permission prevents a user from deleting other users' files from public directories such as /tmp:

|  |
| --- |
| drwxrwxrw**t** 7 root sys 400 Sep 3 13:37 tmp |

Be sure to set the sticky bit manually when you set up a public directory on a TMPFS file system.