

Referencias

<https://deephacking.tech/permisos-sgid-suid-y-sticky-bit-linux/#sgid>

<https://www.redhat.com/sysadmin/suid-sgid-sticky-bit>

Definiciones

- **SUID:**

Aplicado a un fichero, hace que a este se ejecute como si lo hubiera ejecutado el propietario sin importar quien lo haya realmente ejecutado.

No puede ser aplicado a directorios porque el único usuario individual que puede manejar el funcionamiento de directorios es root.

- **SGID:**

Aplicado a un fichero, funciona igual que SUID pero en vez como propietario, el fichero se ejecuta con los permisos de un miembro del grupo.

Aplicado a un directorio, hace que todos los ficheros creados en ese directorio reciban el grupo del directorio (junto con los permisos y la pertenencia que ello implica).

- **Sticky bit:**

Aplicado a un fichero, carga a este en *swap* la primera vez que se ejecute para que la próxima vez lo haga más rápido (pero no parece ser muy usado).

Aplicado a un directorio, impide la eliminación de los ficheros del directorio a todos excepto al propietario de cada fichero (y a root, obviamente).

Ejemplos existentes

```
2024/04/19 (viernes) [18:42:43] anibal@MV-Servidor-Ubuntu
/
$ ls -lF /usr/bin/passwd
-rwsr-xr-x 1 root root 68208 feb  6 12:49 /usr/bin/passwd*

2024/04/19 (viernes) [18:42:55] anibal@MV-Servidor-Ubuntu
/
$ ls -lF /usr/bin/crontab
-rwxr-sr-x 1 root crontab 43720 feb 13  2020 /usr/bin/crontab*

2024/04/19 (viernes) [18:43:36] anibal@MV-Servidor-Ubuntu
/
$ ls -dlF /tmp
drwxrwxrwt 12 root root 4096 abr 19 17:26 /tmp/
```

Para encontrarlos usa el comando *find* así:

```
2024/04/19 (viernes) [18:43:40] anibal@MV-Servidor-Ubuntu
/
$ find . -perm /4000
```

Tienes que cambiar */4000* por el tipo de permiso a buscar. Si quieres más información de cómo funciona el comando en ese modo busca en man *find*. El resumen para lo que nos ocupa es: */4000* para SUID, */2000* para SGID y */6000* para *Sticky bit*.

Ejemplos creados

```
2024/04/19 (viernes) [19:04:05] anibal@MV-Servidor-Ubuntu
~/tests.d
$ ll
total 8
drwxrwxr-x  2 anibal anibal 4096 abr 19 19:04 ./
drwxr-xr-x 12 anibal anibal 4096 abr 19 17:58 ../
-rw-rw-r--  1 anibal anibal   0 abr 19 19:04 file_in_mode_-s-
-rw-rw-r--  1 anibal anibal   0 abr 19 19:03 file_in_mode_s--
-rw-rw-r--  1 anibal anibal   0 abr 19 19:04 file_in_mode_--t

2024/04/19 (viernes) [19:04:08] anibal@MV-Servidor-Ubuntu
~/tests.d
$ chmod 4640 file_in_mode_s--

2024/04/19 (viernes) [19:07:36] anibal@MV-Servidor-Ubuntu
~/tests.d
$ chmod 2640 file_in_mode_-s-

2024/04/19 (viernes) [19:07:47] anibal@MV-Servidor-Ubuntu
~/tests.d
$ chmod 1640 file_in_mode_--t

2024/04/19 (viernes) [19:08:07] anibal@MV-Servidor-Ubuntu
~/tests.d
$ ll
total 8
drwxrwxr-x  2 anibal anibal 4096 abr 19 19:04 ./
drwxr-xr-x 12 anibal anibal 4096 abr 19 17:58 ../
-rw-r-S---  1 anibal anibal   0 abr 19 19:04 file_in_mode_-s-
-rwSr----- 1 anibal anibal   0 abr 19 19:03 file_in_mode_s--
-rw-r-----T 1 anibal anibal   0 abr 19 19:04 file_in_mode_--t
```

Sí, se que el permiso **t** no tiene mucho sentido en un fichero pero esto es un ejemplo teórico, no práctico. Si no lo fuera los demás permisos no serían todos iguales en cada fichero.