SUID on a file

If any class of user is granted execute permission, then this bit causes the `owner` of the resulting process to be that of the file and not of the user running the program. So if the program attempts to `read()` something, the permissions that apply would be for the `owner` of the file and not the user of the program.

For example, suppose user `Jane` runs the command `cat memo.txt`, and the permissions on the `cat` command and the file `memo.txt` are as follows:

```
-rwx--x--x    1 root    bin         4515 Aug 14 13:08 cat
-rw-------    1 root    bin          218 Aug 14 13:08 memo.txt
```

Jane has permission to run `cat`, but not permission to read `memo.txt`. So when this `cat` program attempts to `read()` the file a `permission denied` error will occur.

Suppose we change the cat program to have the `SUID` bit on:

```
-rws--x--x    1 root    bin         4515 Aug 14 13:08 cat
```

Now, when Jane runs this `cat` program, the access to `memo.txt` is permitted. When `cat` attempts to `read()` the file, the system doesn't think `Jane` is attempting to read, it thinks `root` is the user. So the access is allowed.

A similar substitution occurs if the `SGID` bit is set and any execute bits are set. The `group` ID checked is not the current user, but the `group` of the program.