如何使用特殊权限：setuid、setgid 和 sticky 位

内容

目 标

了解特殊权限的工作原理，以及如何识别和设置它们。

要 求

- 了解标准的 Unix / Linux 权限系统 [1]

难 度

简单

约 定

- `#` - 要求直接以 root 用户或使用 `sudo` 命令执行指定的命令
- `$` - 用普通的非特权用户来执行指定的命令

介 绍

通常，在类 Unix 操作系统上，文件和目录的所有权是基于文件创建者的默认 `uid`（user-id）和 `gid`（group-id）的。启动一个进程时也是同样的情况：它以启动它的用户的 `uid` 和 `gid` 运行，并具有相应的权限。这种行为可以通过使用特殊的权限进行改变。

setuid 位

当使用 setuid （设置用户 ID）位时，之前描述的行为会有所变化，所以当一个可执行文件启动时，它不会以启动它的用户的权限运行，而是以该文件所有者的权限运行。所以，如果在一个可执行文件上设置了 setuid 位，并且该文件由 root 拥有，当一个普通用户启动它时，它将以 root 权限运行。显然，如果 setuid 位使用不当的话，会带来潜在的安全风险。

使用 setuid 权限的可执行文件的例子是 passwd，我们可以使用该程序更改登录密码。我们可以通过使用 ls 命令来验证：

```
ls -l /bin/passwd
-rwsr-xr-x. 1 root root 27768 Feb 11 2017 /bin/passwd
```

如何识别 setuid 位呢？相信您在上面命令的输出已经注意到，setuid 位是用 s 来表示的，代替了可执行位的 x。小写的 s 意味着可执行位已经被设置，否则你会看到一个大写的 S。大写的 S 发生于当设置了 setuid 或 setgid 位、但没有设置可执行位 x 时。它用于提醒用户这个矛盾的设置：如果可执行位未设置，则 setuid 和 setgid 位均不起作用。setuid 位对目录没有影响。

setgid 位

与 setuid 位不同，setgid （设置组 ID）位对文件和目录都有影响。在第一个例子中，具有 setgid 位设置的文件在执行时，不是以启动它的用户所属组的权限运行，而是以拥有该文件的组运行。换句话说，进程的 gid 与文件的 gid 相同。

当在一个目录上使用时，setgid 位与一般的行为不同，它使得在所述目录内创建的文件，不属于创建者所属的组，而是属于父目录所属的组。这个功能通常用于文件共享（目录所属组中的所有用户都可以修改文件）。就像 setuid 一样，setgid 位很容易识别（我们用 test 目录举例）：

```
ls -ld test
drwxrwsr-x. 2 egdoc egdoc 4096 Nov 1 17:25 test
```

这次 s 出现在组权限的可执行位上。

sticky 位

sticky （粘连）位的工作方式有所不同：它对文件没有影响，但当它在目录上使用时，所述目录中的所有文件只能由其所有者删除或移动。一个典型的例子是 /tmp 目录，通常系统中的所有用户都对这个目录有写权限。所以，设置 sticky 位使用户不能删除其他用户的文件：

```
$ ls -ld /tmp
drwxrwxrwt. 14 root root 300 Nov 1 16:48 /tmp
```

在上面的例子中，目录所有者、组和其他用户对该目录具有完全的权限（读、写和执行）。sticky 位在可执行位上用 t 来标识。同样，小写的 t 表示可执行权限 x 也被设置了，否则你

会看到一个大写字母 T。

如何设置特殊权限位

就像普通的权限一样，特殊权限位可以用 chmod 命令设置，使用数字或者 ugo/rwx 格式。在前一种情况下，setuid、setgid 和 sticky 位分别由数值 4、2 和 1 表示。例如，如果我们要在目录上设置 setgid 位，我们可以运行：

```
1. $ chmod 2775 test
```

通过这个命令，我们在目录上设置了 setgid 位（由四个数字中的第一个数字标识），并给它的所有者和该目录所属组的所有用户赋予全部权限，对其他用户赋予读和执行的权限（目录上的执行位意味着用户可以 cd 进入该目录或使用 ls 列出其内容）。

另一种设置特殊权限位的方法是使用 ugo/rwx 语法：

```
$ chmod g+s test
```

要将 setuid 位应用于一个文件，我们可以运行：

```
$ chmod u+s file
```

要设置 sticky 位，可运行：

```
$ chmod o+t test
```

在某些情况下，使用特殊权限会非常有用。但如果使用不当，可能会引入严重的漏洞，因此使用之前请三思。