

Extending the File Mode with ACLs



Andrew Mallett

LINUX AUTHOR AND TRAINER

@theurbanpenguin www.theurbanpenguin.com



Module Overview



Limitations of the file mode

Mitigating limitations with POSIX ACLS

Establishing support for ACLs

Viewing and setting ACLs

The default ACL

Adding ACLs to files

Removing ACL and ACL entries





The UNIX file mode was never designed for enterprise file sharing

Allowing for a single user, single group, and everyone else

To work around this, you *can* just keep creating groups to meet new needs in the file system

Even so, this does not cater for when a group requires read access and another group requires read-write access to the same file or directory

ACLs overcome these limitations

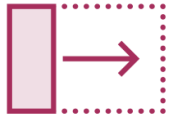


POSIX ACLs

Access Control Lists allow for more than one user or group to have the same or similar permissions to a file resource. We can also set default permissions allowing **new** files or directories to inherit from the parent



Establishing Support for ACLs



```
sudo yum list acl
```



```
grep -i acl /boot/config-$(uname -r)
```



```
rpm -qf $(which getfacl)
```



```
sudo tune2fs -l /dev/sda2 | grep -i acl
```

Demo



We will begin by establishing support for ACLs on our AlmaLinux 8 System



```
$ sudo yum install -y httpd
$ sudo setfacl -m d:u:apache:r,d:o:- /var/www/html
$ sudo echo "Hello" | sudo tee /var/www/html/index.html
$ ls -l /var/www/html/index.html
```

Default ACLs

Default ACLs can be applied only to directories. Useful to ensure services can maintain the correct access to files whilst restricting others. Setting the default ACL on the Apache DocumentRoot will not affect existing content. Create your own new index page and the permissions will be correct



```
$ sudo setfacl -m d:u:apache:r,d:o:- /var/www/html
```

```
setfacl <rules> <files>
```

More than one rule can exist, if so, comma separate them

```
rule 1 d:u:apache:r
```

```
rule 2 d:o:-
```

setfacl

Looking at how we set the ACL in more detail:

-m modifies the ACL -- **d:** in the rule specifies the default ACL

A **user** rule must include the username

A rule for **others** does not contain the username. Here we set no permissions for others



Demo



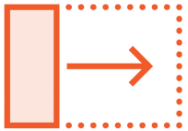
Most useful with ACLs is the default ACL. Keeping practical, so you know WHY we use ACLs, we will use the default ACL to help secure the Apache web server

`setfacl` to set permissions

`getfacl` to view



Using ACLs



As root we can create a private directory

```
$ sudo mkdir -m 700 /private
```



The user vagrant has no access to this but access is required

```
$ sudo setfacl -m u:vagrant:rwx /private
```

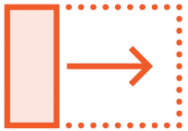


We set the default ACL to allow user access to files created within

```
$ sudo setfacl -m d:u:vagrant:rwx /private
```



Viewing ACLs



Where an ACL is set the standard permissions will display a +
`drwxrwx---+ 2 root root 4096 Mar 24 13:14 /private/`



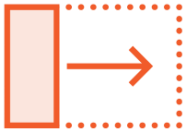
The command `getfacl` reads the ACL
`$ getfacl /private`



To view just the default ACL use **-d**
`$ getfacl -d /private`



Adding and Removing ACL Entries



We can add additional entries if required

```
$ sudo setfacl -m u:tux:r-x /private
```



If that was an error or we simply needed to remove an entry

```
$ setfacl -x u:tux /private
```

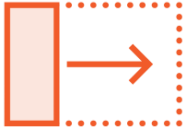


Adding entries for groups is similar to adding users

```
$ setfacl -m g:wheel:r-x /private
```



Backing up and Restoring ACLs



To create a backup of an ACL

```
$ sudo getfacl /private > /tmp/acl.txt
```



Remove the complete ACL

```
$ sudo setfacl -b /private
```



Restore from the / directory. Relative paths are used

```
$ sudo setfacl --restore=/tmp/acl.txt
```



Demo



We will now run through adding and removing ACLs from the command line



Summary

File Mode

Very restrictive with a single user, group and others

ACL

A list of entries we can manage including setting defaults

Tools

The package acl adds the setfacl and the getfacl tools



Implementing SELinux

* * * * *