RHEL: ACLs basics

Article Number: 61 | Rating: Unrated | Last Updated: Sun, May 27, 2018 8:38 PM

Files and directories have permission sets for the owner of the file, the group associated with the file,
and all other users for the system. However, these permission sets have limitations. For example, different
permissions cannot be configured for different users. Thus, Access Control Lists (ACLs) were implemented.

The Red Hat Enterprise Linux kernel provides ACL support for the ext3 file system and NFS-exported file
systems. ACLs are also recognized on ext3 file systems accessed via Samba.

Along with support in the kernel, the acl package is required to implement ACLs. It contains the utilities
used to add, modify, remove, and retrieve ACL information.

The cp and mv commands copy or move any ACLs associated with files and directories.

Mounting File Systems

Before using ACLs for a file or directory, the partition for the file or directory must be mounted with
ACL support. If it is a local ext3 file system, it can be mounted with the following command:

```
# mount -t ext3 -o acl <device-name> <mountpoint>
```

For example:

```
# mount -t ext3 -o acl /dev/VolGroup00/LogVol02 /work
```

If an ext3 file system is accessed via Samba and ACLs have been enabled for it, the ACLs are recognized
because Samba has been compiled with the "**--with-acl-support**" option. No special flags are required when
accessing or mounting a Samba share.

We can use 'tune2fs' to set ACLs as a default mount option:

```
# tune2fs -o acl </dev/volume/home>
```

NFS

By default, if the file system being exported by an NFS server supports ACLs and the NFS client can read
ACLs, ACLs are utilized by the client system.

To disable ACLs on NFS shares when configuring the server, include the "no_acl" option in the /etc/exports
file. To disable ACLs on an NFS share when mounting it on a client, mount it with the "no_acl" option via
the command line or the /etc/fstab file.

Setting ACLs

There are two types of ACLs: access ACLs and default ACLs. An access

ACL is the access control list for
a specific file or directory. A default ACL can only be associated with a directory; if a file within
the directory does not have an access ACL, it uses the rules of the default ACL for the directory.
Default ACLs are optional.

ACLs can be configured:

- Per user
- Per group
- Via the effective rights mask
- For users not in the user group for the file

The 'setfacl' utility sets ACLs for files and directories. Use the "-m" option to add or modify the ACL of a
file or directory:

setfacl -m <rules> <files>

Rules must be specified in the following formats. Multiple rules can be specified in the same command
if they are separated by commas.

u:uid:perms - Sets the access ACL for a user. The user name or UID may be specified. The user may
be any valid user on the system.

g:gid:perms - Sets the access ACL for a group. The group name or GID may be specified. The group
may be any valid group on the system.

m:perms - Sets the effective rights mask (nomal permissions). The mask is the union of all permissions
of the owning group and all of the user and group entries.

o:perms - Sets the access ACL for users other than the ones in the

group for the file.

Permissions must be a combination of the characters "r", "w", and "x" for read, write, and execute.
If a file or directory already has an ACL, and the 'setfacl' command is used, the additional rules are
added to the existing ACL or the existing rule is modified.

For example, to give read and write permissions to user "andrius":

```
# setfacl -m u:andrius:rw /project/somefile
```

To remove all the permissions for a user, group, or others, use the "-x" option and do not specify any
permission:

```
# setfacl -x <rules> <files>
```

For example, to remove all permissions from the user with UID 500:

```
# setfacl -x u:500 /project/somefile
```

Setting Default ACLs

To set a default ACL, add "d:" before the rule and specify a directory instead of a file name.
For example, to set the default ACL for the "/share" directory to read and execute for users not in
the user group (an access ACL for an individual file can override it):

```
# setfacl -m d:o:rx /share
```

Retrieving ACLs

To determine the existing ACLs for a file or directory, use the 'getfacl' command. In the example below,
the 'getfacl' is used to determine the existing ACLs for a file.

```
# getfacl  home/john/picture.png
```

The above command returns the following output:

```
# file: home/john/picture.png
# owner: john
# group: john
user::rw-
group::r--
other::r--
```

If a directory with a default ACL is specified, the default ACL is also displayed as illustrated below.
For example, "**getfacl home/sales/**" will display similar output:

```
# file: home/sales/
# owner: john
# group: john
user::rw-
user:barryg:r--
group::r--
mask::r--
other::r--
default:user::rwx
default:user:john:rwx
default:group::r-x
```

```
   default:mask::rwx
   default:other::r-x
```

Archiving File Systems with ACLs

By default, the 'dump' command now preserves ACLs during a backup operation. When archiving a file or file system with 'tar', use the "--acls" option to preserve ACLs. Similarly, when using 'cp' to copy files with ACLs, include the "--preserve=mode" option to ensure that ACLs are copied across too. In addition, the "-a" option (equivalent to "-dR --preserve=all") of 'cp' also preserves ACLs during a backup along with other information such as timestamps, SELinux contexts, and the like.

The 'star' utility is similar to the 'tar' utility in that it can be used to generate archives of files; however, some of its options are different. The 'star' package is required to use this utility.

Compatibility with older systems

If an ACL has been set on any file on a given file system, that file system has the "ext_attr attribute. This attribute can be seen using the following command:

```
   # tune2fs  -l  <filesystem-device>
```

A file system that has acquired the "ext_attr" attribute can be mounted with older kernels, but those
kernels do not enforce any ACLs which have been set.

Versions of the 'e2fsck' utility included in version 1.22 and higher of the 'e2fsprogs' package (including
the versions in Red Hat Enterprise Linux 2.1 and 4) can check a file system with the "ext_attr" attribute.
Older versions refuse to check it.