**Access Control Lists in Linux**
***Author - Nic Maurel***

Many linux users today have become familiar with using the standard posix format, but it was argued that the standard *user/group/other* permissions do not provide enough granularity with regards to assigning permissions. What if I told you it was possible to assign more than one user or more than one group to a file to a directory or file. Interested? Lets continue. The good news is that acl support comes by default in the 2.6 Kernel (checked on Fedora 2 and 4), some 2.4 kernels would have to be patched. The Linux file systems ext2fs, ext3fs, reiserfs, jfs, xfs, and nfs can all be ACL enabled, this depends on the kernel. Now I am not going to talk you through patching the kernel but merely on how to check whether you have acl support enabled.

*Checking*
On *fedora Core 4* search for this file

```
#vi /usr/src/kernels/2.6.xxxxxxversion/.config
```

and search for these options

```
CONFIG_EXT2_FS_POSIX_ACL=y
CONFIG_EXT3_FS_POSIX_ACL=y         <----  As you can see we have built in support
CONFIG_REISERFS_FS_POSIX_ACL=y
CONFIG_JFS_POSIX_ACL=y
CONFIG_FS_POSIX_ACL=y
CONFIG_XFS_POSIX_ACL=y
CONFIG_NFS_ACL=y
CONFIG_NFSD_ACL=y
CONFIG_NFS_ACL_SUPPORT=y
```

If you are wanting support with samba type this

```
# smbd -b | grep ACL
  HAVE_SYS_ACL_H
  HAVE_POSIX_ACLS
```

Once again by default on Fedora Core 4's samba version has ACL's built in. Now the last thing you need to do is mount the file system with ACL support

```
# mount
/dev/hda1 on / type ext3 (rw)
/dev/hda3 on /data02 type ext3 (rw)
```

I have filtered the output to the specific filesystems that we would look at and as you can see the filesystem is not mounted with acl support so lets do it!

```
# mount / -o remount,acl
```

This can be done seamlessly without affecting normal operation, but if you are not sure rather be safe then sorry. Right lets check now.

```
# mount
/dev/hda1 on / type ext3 (rw,acl)
/dev/hda3 on /data02 type ext3 (rw)
```

Cool! We now have acl support and we are ready to start applying permissions

*Using it*
There are two commands you need to know to read,edit and add acl permissions.

- *getfacl* is the first one, self explanatory, it reads the acl attributes
- *setfacl* is the second, sets edits and removes permissions, this one is a little more complicated

Lets take a look at *setfacl*..

```
setfacl options  [user]or[group]:[uid][gid]or[username][groupname]:[Octalpermissions]or[rwxPermissions] ...
filename
```

looks like theres a lot there but there really isn't, lets take a look at an example..

```
setfacl -m u:root:rwx file
or
setfacl -m user:0:7 file
```

they both do the same thing

- *-m* : modify acl attributes from file or directory
- *-x :* removes acl attributes from file or directory
- *-b :* removes all acl attributes from file or directory
- *-R* : recursive used in conjunction with m,x or b
- *--restore-file=filename :* restores acls explained later

```
[root ]# useradd test
[root ]# cd /home/test
[root ]# touch testfile ; echo hello > testfile
[root ]# ls -l testfile
-rw-r--r--  1 root root 9 Apr 13 08:51 testfile
```

lets modify these permissions so that only root has access to this file..

```
[root ]# chmod 770 testfile
[root ]# ls -l testfile
-rwxrwx---   1 root      root          9 Apr 13 08:51 testfile
```

And there we have it root is the only user that can access testfile. Now lets apply an acl attribute..

```
[root ]# setfacl -m u:test:rwx testfile
[root ]# ls -l testfile
-rwxrwx---+ 1 root root 6 Apr 13 08:51 testfile
```

You will now notice that there is a + sign to symbolize extended attributes, lets now read it ..

```
[root ]# getfacl testfile
# file: testfile
# owner: root
# group: root
user::rwx
user:test:rwx <----- user test has effective permissions of rwx
group::rwx
mask::rwx
other::---

[root ]# cat testfile
hello         <----- remember we did this at the beginning
[root ]# su test
[test ]$ echo "test says hello" >> testfile
[test ]$ cat testfile
hello
test says hello
```

And there we have it acls in action, quite scary! When you think about it. There are a few more things you need to know;

- Standard tar and gzip will not backup these acls as the format cannot be recognized

so here is a quick tip on how to backup and restore those acls..

```
getfacl -R --skip-base / > /backup.acl
```

This backs up the acls of file that only have acls (skips the base permission files) on your root partition to a backup.acl file. Got that? shew!

```
# cd /
# setfacl --restore=backup.acl
```

And that's pretty much it, sit back and let it restore the acls.

*Final Words*

There is a lot more to be learnt about acls under linux, eg. masks, editing file permissions on samba through ntacl control panel, modifying the permissions in general. I just thought this would be a quick guide to get people started on using acls. Remember one thing always test things first before using the shotgun approach on a production box as this could get you into serious trouble as well as allow access to users that don't require it (least priviledge). I hope you had fun!