LEARN. NETWORK. EXPERIENCE OPEN SOURCE.

June 11-14, 2013
Boston, MA

redhat.

Filesystem Access Control Lists

Rudolf Kastl
Curriculum Manager, Red Hat
June 14th, 2013

ACLs – Overview

- This session will cover advanced file system permission features which may be used when controlling access to files and directories:
  - File system Access Control Lists

ACLs – Access Control Lists

- Access Control Lists allow more complex file access permissions to be expressed:

  “On this file, Paul should have read-write, Laura read-only, group **finance** read-write, group **audit** read-only, and nobody else should have access”

- The **ls -l** command will display a **+** if a file has an ACL:

```
-rw-rw-r--+  1 rkastl rkastl      8856 May 11 16:53 file
```

ACLs – Commands

- **setfacl** sets an ACL entry on a file..

  setfacl -m u:<username>:<perms> <filename>

  setfacl -m g:<groupname>:<perms> <filename>

  ...or removes it:

  setfacl -x u:<username> <filename>

  setfacl -x g:<groupname> <filename>

- **getfacl** gets the list of ACL entries on a file

  getfacl <filename>

ACLs – Example

- **setfacl** sets an ACL entry on a file..

```
[root@desktopX ~]# echo hello > /tmp/test.txt
[root@desktopX ~]# setfacl -m u:student:rw /tmp/test.txt
[root@desktopX ~]# getfacl /tmp/test.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/test.txt
# owner: root
# group: root
user::rw-
user:student:rw-
group::r--
mask::rw-
other::r--
```

ACLs – Example

- Or removes the ACL entry...

```
[root@desktopX ~]# setfacl -x u:student /tmp/test.txt
[root@desktopX ~]# getfacl /tmp/test.txt
getfacl: Removing leading '/' from absolute path names
# file: tmp/test.txt
# owner: root
# group: root
user::rw-
group::r--
mask::rw-
other::r--
```

ACLs – Precedence of permissions

- 4 easy steps, first match takes effect and we stop:

Step	Question	Permission
1	Is the process running as the user that is owner of the file?	**User** permissions apply
2	Does the file have an ACL entry set for the process's user?	**User's ACL entry** applies
3	Is the process running as • the group that owns the file, or • a group that has a group ACL entry?	**Any matching Group or Group ACL entry** granting access applies
4	Do none of the above match?	Permissions for **other** applies

ACLs – Mask

- The **ACL mask** determines the maximum permissions for the group that owns the file and any users or groups ACL entries
- Effective rights are shown by the **getfacl** command

ACLs – Mask Example

- Using the mask for revoking permissions...

```
[root@desktopX ~]# touch /tmp/script.sh
[root@desktopX ~]# setfacl -m u:student:rwx /tmp/script.sh
[root@desktopX ~]# setfacl -m m::rx /tmp/script.sh
[root@desktopX ~]# getfacl /tmp/script.sh
getfacl: Removing leading '/' from absolute path names
# file: tmp/script.sh
# owner: root
# group: root
user::rw-
user:student:rwx        #effective:r-x
group::r--
mask::r-x
other::r--
```

ACLs – Mask

- The group permissions shown by **ls -l** represent the mask, not the actual owning group's permissions on a file with ACLs.
- A **chmod** command which tries to alter the group permission effectively alters the mask.

ACLs – Mask

- The mask gets recalculated with every new ACL entry set or modified unless the **-n** switch is used, so mask restrictions have to be reapplied.
  - **setfacl -n -m group::rwx script.sh**
    Sets group permissions to read write and execute without recalculating the mask

ACLs – Inheritance (Default ACLs)

- Default ACLs on a directory allow you to automatically set ACL entries on files created in that directory:

setfacl -m d:u:<user>:<perms> <directory>
setfacl -m d:g:<group>:<perms> <directory>

ACLs – Inheritance Example

- Lets create a directory and set an inheritance ACL...

```
[root@desktopX ~]# mkdir /tmp/testdir
[root@desktopX ~]# setfacl -m d:u:student:rwx /tmp/testdir
[root@desktopX ~]# getfacl /tmp/testdir
getfacl: Removing leading '/' from absolute path names
# file: tmp/testdir
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:user:student:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

ACLs – Inheritance Example

- With a regular file the execute bit of the inheritance rule gets stripped with the mask...

```
[root@desktopX ~]# touch /tmp/testdir/testfile
[root@desktopX ~]# getfacl /tmp/testdir/testfile
getfacl: Removing leading '/' from absolute path names
# file: tmp/testdir/testfile
# owner: root
# group: root
user::rw-
user:student:rwx    #effective:rw-
group::r-x          #effective:r--
mask::rw-
other::r--
```

ACLs – Inheritance Example

- A subdirectory behaves as expected...

```
[root@desktopX ~]# mkdir /tmp/testdir/subdir
[root@desktopX ~]# getfacl /tmp/testdir/subdir
getfacl: Removing leading '/' from absolute path names
# file: tmp/testdir/subdir
# owner: root
# group: root
user::rwx
user:student:rwx
group::r-x
mask::rwx
other::r-x
default:user::rwx
default:user:student:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

Filesystem ACLs – Exercise

- A Lab exercise has been provided so you can practice using Filesystem ACLs
- Please let me know if you have any questions

Thank you for attending!

WANT TO LEARN MORE ABOUT FILESYSTEM PERMISSIONS?

- Learn more about our course offerings at

www.redhat.com/training