Filename: comptia-linuxxk0004-3-2-1-advanced_permissions
Show Name: CompTIA Linux+ (XK0-004)
Topic: Managing Permissions and Ownership
Episode Name: Advanced Permissions
Description: In this episode, Zach and Don continue their permissions adventure. This time they show how to overcome the limitations of UNIX style permissions through the use of file system access control lists or FACLs. They demonstrate using the setfacl and getfacl commands to manage permissions with FACLs.

Advanced Permissions

[?] What limitations do we have with basic permissions?

- Limitations
  - One user
  - One group
  - Users apply their primary group by default
  - Not very flexible
- Create users
  - `useradd <username>`
  - `passwd <username>`
- Create groups:
  - `groupadd <groupname>`
  - `gpasswd -a <user> <group>`

[?] So, how can we overcome those limitations?

- Filesystem Access Control Lists
  - ACLs/FACLs
  - Allow assigning permissions to more than one user/group

[?] Are FACLs available in all distributions?

- Must be enabled with the `acl` option in `/etc/fstab`
  - `tune2fs -l /dev/vg1/website | grep "mount options"`
- RHEL enables them by default during installation

[?] How do we actually configure the FACLs?

- Viewing the current ACL
  - `getfacl <file>`
  - If an ACL is present, there will be a `+` in the permissions list
    - `drwxr-x-r-x+`
- Assigning permissions for a user
  - `setfacl -m u:<username>:rwx <filename>`
  - `setfacl -s u::rwx,g::r-x,o::r-x,u:<username>:rwx <filename>`
  - `-m` modifies
  - `-s` sets (replaces)
  - `-x` removes

[?] Are group permissions handled the same way?

- Assigning permissions for a group
    - `setfacl -m g:<group>:rwx <filename>`
    - Group permissions cannot exceed those assigned to the default group
        - `rwxr-xr-w` would not allow write
        - This is called a *mask*

[?] Do FACLs work with directories as well?

- Directory permissions can be passed down to files
    - Inheritance
    - Must be enabled
    - `setfacl -m d:g:<group>:rwx <directory>`
    - The `d:` indicates it is a directory default
    - `f:` can be used to set a file default

[?] Are there any other advanced permissions we should be aware of?

- Sticky bit
    - Files with the sticky bit can only be deleted/renamed by *root* or the file owner.
    - Allows for "Friendly" shared directories
    - `chmod o+t <directory>`
    - Shows as a "t"
        - `rwxr-xr-t`