Red Hat System Administration I

UNIT 6

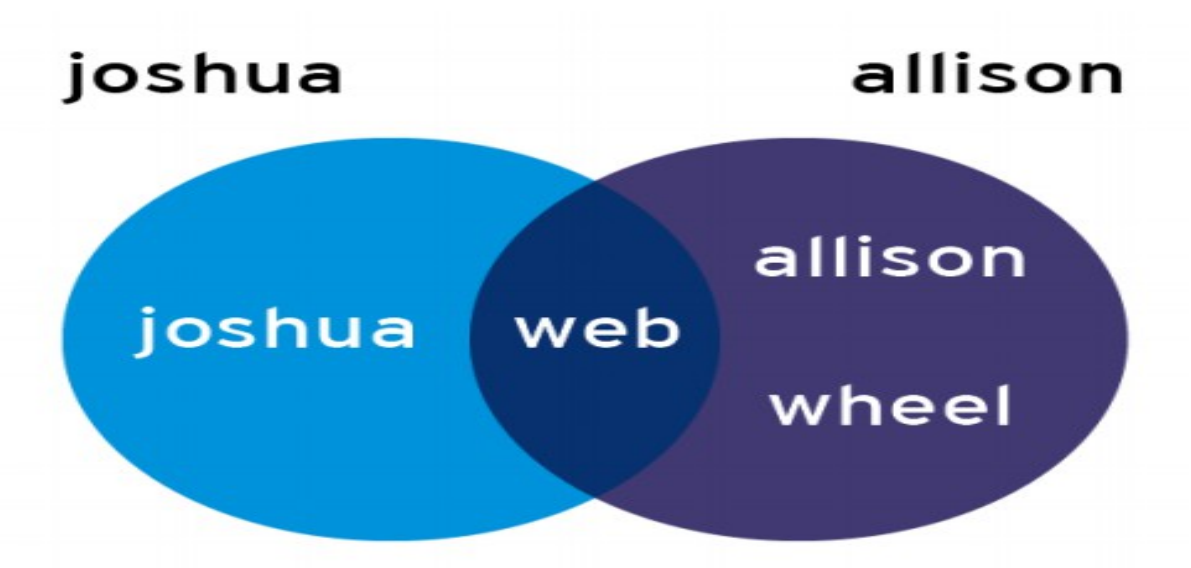
Controlling Access to Files with Linux File System Permissions

Objectives

- Explain how the Linux file permissions model works.
- Change the permissions and ownership of files using command-line tools.
- Configure a directory in which newly created files are automatically writable by members of
- the group which owns the directory, using special permissions and default umask settings.

Linux File System Permissions

- Access to files by users are controlled by filepermissions



Effects of permissions on files and directories

Permission	Effect on files	Effect on directories
r (read)	Contents of the file can be read.	Contents of the directory (file names) can be listed.
w (write)	Contents of the file can be changed.	Any file in the directory may be created or deleted.
x (exec)	Files can be executed as commands.	Contents of the directory can be accessed (dependent on the permissions of the files in the directory).

Viewing file/directory permissions and ownership

- **For file**

  [student@desktopX ~]$ls -l test
  -rw-rw-r--. 1 student student 0 Feb 8 17:36 test

- **For direcotry**

  [student@desktopX ~]$ls -ld /home
  drwxr-xr-x. 5 root root 4096 Jan 31 22:00 /home

Managing File System Permissions from the Command Line

- How to chang file|directory permissions
  - chmod [Who][What][Which] file|directory

    Whois u, g, o, a (foruser,group,other,all)

    Whatis +, -, = (foradd,remove,setexactly)

    Whichis r, w, x (forread,write,executable)
  - chmod ### file|directory

    # is sum of r=4, w=2, and x=1

Changing file/directory user or group ownership

- How to change file|directory user
  - chown user directory
- How to change file|directory group
  - chgroup [-R] group directory

Special permissions

- **Suid,Sgid,Stickyid**

Special permission	Effect on files	Effect on directories
u+s (suid)	File executes as the user that owns the file, not the user that ran the file.	No effect.
g+s (sgid)	File executes as the group that owns the file.	Files newly created in the directory have their group owner set to match the group owner of the directory.
o+t(sticky)	No effect.	Users with **write** on the directory can only remove files that they own; they cannot remove or force saves to files owned by other users.

Setting special permissions

- Symbolically: setuid = u+s; setgid = g+s; sticky = o+t
- Numerically (fourth preceding digit): setuid = 4; setgid = 2; sticky = 1

Default file permissions

- What is umask
  - Directory or file to remove permissions at creation time
- How to set umask
  - umask permission's number
  - vim /etc/bashrc and vim /etc/profile

Lab

<lab 1>
Open a terminal window and become root on serverX.<>
<lab 2>
Create the /home/stooges directory
<lab 3>
Change group permissions on the /home/stooges directory so it belongs to the stooges group
<lab 4>
Set permissions on the /home/stooges directory so it is a set GID bit directory (2), the owner (7) and group (7) have full read/write/execute permissions, and other users have no permission (0) to the directory
<lab 5>
Check that the permissions were set properly
<lab 6>
Modify the global login scripts so that normal users have a umask setting which prevents others from viewing or modifying new files and directories.
<lab 7>
When you finish, open a terminal window on serverX and run lab permissions grade to confirm you have done everything correctly.

That's all