Capgemini

UNIX

Lesson 07 : Access Permissions

Lesson Objectives

- In this lesson, you will learn:
  - File permissions
  - Changing permission
  - Changing owenership

7.1: File permission
Access Permissions

- UNIX is a multi-user system. Every file and directory in your account can be protected from or made accessible to other users by changing its access permissions. Every user has responsibility for controlling access to their files.
- Permissions for a file or directory may be any or all of:
  - r - read
  - w - write
  - x - execute = running a program
- Each permission (rwx) can be controlled at three levels:
  - u - user = yourself
  - g - group = can be people in the same project
  - o - other = everyone on the system

Access Permissions

File access permissions are displayed using the ls -l command.

The output from the ls -l command shows all permissions for all levels as three groups of three according to the scheme:

- owner read (r) owner write (w) owner execute (x)
- group read (r) group write (w) group execute (x)
- public read (r) public write (w) public execute (x)
- which are displayed as: -rwxrwxrwx
- Note: a directory must have both r and x permissions if the files it contains are to be accessed.

Access Permissions

The chmod command is used to change access permissions for files which you own. The syntax is:

chmod permission_triads filename

[who][action][permissions]

where: **who action permissions**

u = user + = add r = read

g = group - = remove w = write

o = other x = execute

a = all

- chmod a+r sample.f - *Adds read permission for all users to the file sample.f.*
- chmod o-r sample.f - *Removes read permission for others to the file sample.f.*
- chmod og+rx prog* - Adds read and execute permissions for group and others to all files which contain "prog" as the first four characters of their name.
- chmod +w * - Adds write permission for user to all files in current directory.

Access Permissions

File access permissions can also be changed by a numerical (octal) chmod specification. Read permission is given the value 4, write permission the value 2 and execute permission 1.

r w x

4 2 1

These values are added together for any one user category:

0 = no permissions

1 = execute only

2 = write only

3 = write and execute (1+2)

4 = read only

5 = read and execute (4+1)

6 = read and write (4+2)

7 = read and write and execute (4+2+1)

Access Permissions

So access permissions can be expressed as three digits.

For example:

	user	group	others
chmod 640 file1	**rw-**	**r--**	**---**
chmod 754 file1	**rwx**	**r-x**	**r–**
chmod 664 file1	**rw-**	**rw-**	**r–**

Never set write permission for all other users on a file or directory which is in your home directory. If you do other users will be able to change its content. This can represent a serious security risk.

7.2: Changing ownership

Change file ownership

- chown
  - Can be issued by owner of file ONLY.
  - Syntax: chown newowner filename
  - Ownership once given cannot be revoked.

SUMMARY

- Read, write and execute permission on files
- Use of Chmod and chown command

Review Questions

- ____ command is used to change ownership of file?
- _____ can only change owner of file?
- How to assign rwx permission to owner ,rw- permission to other and group?
  - Chmod a=rwx file
  - Chmod u=rwx,go=rw file
  - Chmod 766 file
  - Chmod 667 file