RHCEv8 Online Class 15052021 10:00pm
RHCSA-sysadmin
- Controlling Access to Files
- Controlling Access to Files with ACLs

- Controlling Access to Files
LINUX FILE-SYSTEM PERMISSIONS
what is permission?
who has access and how much to file/dir and who doesn't have access.

ex:
touch /tmp/file1
mkdir /tmp/coss
-file
ls -l /tmp/file1
-rw-r--r--. 1 root root 0 May 15 10:06 /tmp/file1
ll -l /tmp/file1
-rw-r--r--. 1 root root 0 May 15 10:06 /tmp/file1
-dir
ls -ld /tmp/coss/
drwxr-xr-x. 2 root root 6 May 15 10:06 /tmp/coss/
ll -ld /tmp/coss/
drwxr-xr-x. 2 root root 6 May 15 10:06 /tmp/coss/

- **->object type**
- >file
d ->dir
b ->block device(storage)
l ->soft link
c ->named character
p ->named pipe
s ->socket

--- --- --- ->permission

. **->type of security**
. ->SELinux(def.)
+ ->Access Control List-ACL (Adv. perm)

1 **->number of hard link**

root ->userowner

root ->userowner's primary group

0 ->size

MAY 15 10:06 ->timestamp

/tmp/file1 ->object name

permission on Linux
read ->r ->4
write ->w ->2
execute ->x ->1

7
user ->u ->userowner
group ->g ->groupowner
other ->o ->who is not userowner either member of groupowner becomes to **other**

u g o
rwx rwx rwx

-read permission ->cat
-write permission ->vim, >, >>
-execute permission ->just available for directory and means get inside directory by use ‘cd’ command
e:
ll -l /tmp/file1
-rw-r--r--
ls -ld /bin/
dr-xr-xr-x

change permission
1-symbolic
2-numeric

chmod <new perm.> file/dir

1-symbolic
r ->read
w ->write
x ->execute
u ->user
g ->group
o ->other
a ->all
+ ->unary plus
- ->unary mines
= ->equal

2-numeric
0
1
2
3
4
5
6
7

ex: numbers
ls -l /tmp/file1
-rw-r--r-- ->644
chmod 646 /tmp/file1
ls -l /tmp/file1
-rw-r--rw- ->646
ex: symbols
ls -l /tmp/file1
-rw--w--w-
chmod g+x /tmp/file1
chown g+r,o+r /tmp/file1
chmod g+r,o+r /tmp/file1
chmod a-w /tmp/file1

# su - user1
$ cat /tmp/file1
$ echo "user1" > /tmp/file1
$ exit
or
su - user1 -c 'cat /tmp/file1'
su - user1 -c 'echo "user11" >> /tmp/file1'

NOTE:
-file full permission ->**666**
-directory full permission ->**777**

default permission on Linux
-root file 644
dir 755

-user file 664
dir 775

how Linux knows about default permission?
user mask-umask
0 0 0 0
s.p u g o
umask
0022
$ umask
0002

-root	file	dir
full perm.	666-	777-
umask	0022	0022
	---	---
def. perm.	0644	0755

-user	file	dir
full perm.	666-	777-
umask	0002	0002
	---	---
def. perm.	0664	0775

How to change default permission on Linux?
step1-custom new permission for file
u g o
rw- r-- r-- ->default permission
-w- rw- -w- ->new permission
2 6 2 ->permission on number
file full perm. 666-
custom perm. 262

0404 ->new umask
step2-implement new mask on Linux
1-temporary
umask 0404
umask
0404
touch /tmp/testfile
ls -l /tmp/testfile
--w-rw--w-
mkdir /tmp/Jupiter
ls -ld /tmp/jupiter/
d-wxrwx-wx
2-persistently
vim /etc/bashrc
line75 umask 404
:wq!
source /etc/bashrc
or
bash

How to change user/group ownership?
-userowner
ls -l /tmp/file2
-rw-r--r--. 1 root root
# id devops
uid=1000(devops) gid=1000(devops) groups=1000(devops)
# chown devops /tmp/file2
# ls -l /tmp/file2
-rw-r--r--. 1 devops root

-groupowner
ls -l /tmp/file2
-rw-r--r--. 1 devops root
groupadd nasa
groupadd India
chgrp nasa /tmp/file2
ls -l /tmp/file2
-rw-r--r--. 1 devops nasa
or
chown **:**India /tmp/moon/
ls -ld /tmp/moon/
drwxrwxr-x. 2 user1 India

-user/group owner at same time
chown <userowner>:<groupowner> <file/dir>
chown user2:devops /tmp/ibm/
ls -ld /tmp/ibm/
drwxr-xr-x. 2 user2 devops

without switch to users, run commands
su - user1 -c 'cat /tmp/file1'

create directory with specific permission
mkdir -m 777 /mnt/dir1
ls -ld /mnt/dir1/
drwxrwxrwx

create file with specific permission
install /dev/null -m 666 /mnt/file12
ls -l /mnt/file12
-rw-rw-rw-

change permission for dir and whole object inside it
mkdir /mnt/testdir
touch /mnt/testdir/file{1..6}
ls -ld /mnt/testdir
-rwxr-xr-x
ll /mnt/testdir/
chmod -R 777 /mnt/testdir/
-R, recursive. means dir + whatever available inside it
ls -ld /mnt/testdir/
drwxrwxrwx
ll /mnt/testdir/

change user/group ownership for dir and whole object inside it
chown -R user2:India /mnt/testdir/

create file with fake size

```
# dd if=/dev/zero of=/media/file1 bs=1M count=100
# du -h /media/file1
100M    /media/file1
```

Special Permission
umask
0022 **0** 0 0 0
s.p
3special permissions
1-set user-id (set uid) ->u->4
2-set group-id (set gid) ->g->2
3-sticky-bit ->t->1

1-set user-id (set uid) ->u->4
-just set it over executable binary commands (Linux commands).
if want users current/futures be able to run command likes userowner, set **set-uid** permission over that command.
ex:
which fdisk
/usr/**sbin**/fdisk
ls -l /usr/sbin/fdisk
-rwxr-xr-x. 1 **root root**
su - user1 -c "fdisk /dev/sdb"
fdisk: cannot open /dev/sdb: Permission denied
-set
chmod 4755 /usr/sbin/fdisk
or
chmod u+s /usr/sbin/fdisk
ls -l /usr/sbin/fdisk
-rwsr-xr-x. 1 root root
-unset
chmod 0755 /usr/sbin/fdisk
or
chmod u-s /usr/sbin/fdisk
chmod 0755 /usr/sbin/fdisk
ls -l /usr/sbin/fdisk
-rwxr-xr-x. 1 root root
su - user1 -c "fdisk /dev/sdb"
fdisk: cannot open /dev/sdb: Permission denied

2-set group-id (set gid) ->g->2
-just set it over directory
set **get-id** over parent directory, objects will create inside parent directory get inherit from parent directory group ownership.
ex:
groupadd India
mkdir /media/mars
ll /media/
drwxr-xr-x. 2 root root 6 May 15 12:01 mars
chgrp India /media/mars/
ll /media/
drwxr-xr-x. 2 root India 6 May 15 12:01 mars
touch /media/mars/f1
ls -l /media/mars/f1
-rw-r--r--. 1 root root
-set
chmod g+s /media/mars/
or
chmod 2755 /media/mars/
ll -ld /media/mars
drwxr-sr-x. 2 root India 26 May 15 12:07 /media/mars
touch /media/mars/f2
ls -l /media/mars/f2
-rw-r--r--. 1 root India

-unset
chmod g-s /media/mars/
ls -ld /media/mars/
drwxr-xr-x. 2 root India 26 May 15 12:07 /media/mars/
touch /media/mars/f3
ls -l /media/mars/f3
-rw-r--r--. 1 root root

3-sticky-bit ->t->1
-just set it over directory
-to prevent suddenly delete files inside directory, set it over parent directory.
ex:
# useradd user10
# useradd user11
# su - user10
$ mkdir /tmp/user10
$ touch /tmp/user10/file{1..5}
$ chmod -R 777 /tmp/user10/
$ ls -ld /tmp/user10/
drwxrwxrwx
$ ll /tmp/user10/
$ exit
# su - user11
$ echo "uer11" >/tmp/user10/file1
$ cat /tmp/user10/file1
uer11
$ rm -rf /tmp/user10/file1
$ exit
# su - user10
-set
$ chmod o+t /tmp/user10/
or
$ chmod 1777 /tmp/user10/
$ ls -ld /tmp/user10/
drwxrwxrwt
$ exit
# su - user11
$ echo "uer11" >/tmp/user10/file2
$ cat /tmp/user10/file2
uer11
$ rm -rf /tmp/user10/file2
rm: cannot remove '/tmp/user10/file2': Operation not permitted
$ exit
# su - user10
-unset
$ chmod o-t /tmp/user10/
or
$ chmod 0777 /tmp/user10/

SuperUserDo-sudo user
-transfer root privileges to other users, solution is sudo
How it works
visudo
or
vim /etc/sudoers
user1 servera.lab.example.com=(root) ALL
user1 ALL=(ALL) ALL
user1 ->sudo user
servera.lab.example.com ->target host
(root) ->target user
ALL ->commands

ex:
useradd user1
passwd user
vim /etc/sudoers
user1 ALL=(ALL) ALL
:wq!
# su - user1
$ which useradd
/usr/sbin/useradd
$ useradd user20
useradd: Permission denied.
$ sudo useradd user20
[sudo] password for user1: redhat
$ id user20
$ sudo fdisk /dev/sdb
$ sudo userdel -rf user2

filter commands
vim /etc/sudoers
user1 ALL=(ALL) /usr/sbin/fdisk**,**/usr/sbin/userdel
:wq!
# su - user1
$ sudo useradd user21
Sorry, user user1 is not allowed to execute '/sbin/useradd user21' as root on servera.lab.example.com.
$ sudo userdel -rf user4
$
$ sudo fdisk /dev/sdb

prevent user to run commands through sudo
vim /etc/sudoers
user1 ALL=(ALL) ALL,**!**/usr/sbin/fdisk**,!**/usr/sbin/userdel
:wq!
# su - user1
$ sudo fdisk /dev/sdb
Sorry, user user1 is not allowed to execute '/sbin/fdisk /dev/sdb' as root on servera.lab.example.com.
$ sudo userdel -rf user5
Sorry, user user1 is not allowed to execute '/sbin/userdel -rf user5' as root on servera.lab.example.com.
$ sudo useradd user6
$

put sudo files in /etc/sudoers.d/ directory
vim /etc/sudoers.d/user1
user1 ALL=(ALL) ALL,!/usr/sbin/fdisk,!/usr/sbin/userdel
:wq!

sudo on group

convert multiple users as sudo users

-create a group

-add them to group as secondary member

-add group in to sudo file

ex:

```
# groupadd testgrp
# useradd user30
# useradd user31
# useradd user32
# echo "redhat" | passwd --stdin user30
# echo "redhat" | passwd --stdin user31
# echo "redhat" | passwd --stdin user32
# gpasswd -a user30 testgrp
# gpasswd -a user31 testgrp
# gpasswd -a user32 testgrp
# groupmems -lg testgrp
user30  user31  user32
# grep "testgrp" /etc/group
testgrp:x:1014:user30,user31,user32
# vim /etc/sudoers
%testgrp        ALL=(ALL) ALL
:wq!
# su - user32
$ sudo useradd user42
[sudo] password for user32:
$
$ exit
```

what is sudo default group

group name is **wheel**

ex:

```
# useradd user42
# passwd user42
# gpasswd -a user42 wheel
# grep "wheel" /etc/group
wheel:x:10:user42
# su - user42
$ sudo useradd user43
[sudo] password for user42:
$
```

run sudo commands without password

```
# vim /etc/sudoers
root            ALL=(ALL)       NOPASSWD: ALL
%testgrp        ALL=(ALL)       NOPASSWD: ALL
:wq!
# su - user30
$ sudo useradd user44
$
```

Access Control List-ACL
helps us and allow to set permission for specific user/group
ex:
touch /tmp/a.txt
ls -l /tmp/a.txt
-rw-r--**r**--. 1 root root
now if permission will change, it will change suppose for whole others users.
chmod 646 /tmp/a.txt
ls -l /tmp/a.txt
-rw-r--**rw**-

-get ACL
getfacl /tmp/a.txt
-set ACL
chmod 640 /tmp/a.txt
ls -l /tmp/a.txt
-rw-r-----
setfacl -m u:<username>:rwx**,**g:<groupname>:rwx,o::rws <file/dir>
setfacl -m u:user20:r /tmp/a.txt
getfacl /tmp/a.txt
user:user20:r--
-verify
su - user1 -c 'cat /tmp/a.txt'
cat: /tmp/a.txt: Permission denied
su - user42 -c 'cat /tmp/a.txt'
cat: /tmp/a.txt: Permission denied
su - user20 -c 'cat /tmp/a.txt'
#
setfacl -m u:user30:r,u:user42:rw,g:nasa:rw /tmp/a.txt
getfacl /tmp/a.txt
user::rw-
user:user20:r--
user:user30:r--
user:user42:rw-
group::r--
group:nasa:rw-
mask::rw-
other::---
su - user42 -c "echo "user42" >/tmp/a.txt"
su - user20 -c "cat /tmp/a.txt"
user42
su - user30 -c "cat /tmp/a.txt"
user42
su - user30 -c "echo "user30" >/tmp/a.txt"
-bash: /tmp/a.txt: Permission denied
copy ACL from one file to other file
getfacl /tmp/a.txt
touch /media/b.txt
getfacl /tmp/a.txt | setfacl --set-file=- /media/b.txt
getfacl /media/b.txt
user::rw-
user:user20:r--
user:user30:r--
user:user42:rw-
group::r--
group:nasa:rw-
mask::rw-
other::---

remove ACL

-user

setfacl -x u:user30 /media/b.txt

-group

setfacl -x g:nasa /media/b.txt

send file back to SELinux

setfacl -b /media/b.txt

touch /media/c.txt

ls -l /media/c.txt

-rw-r--r--**.** 1 root root 0 May 15 13:17 /media/c.txt

chmod 640 /media/c.txt

ls -l /media/c.txt

-rw-r-----**.**

setfacl -m u:user1:r /media/c.txt

ls -l /media/c.txt

-rw-r-----**+**