IT2654

Systems Administration & Security

Topic 4: ACL & Backup

Objectives

- Manage file security by ACL
- Backup & Restore using:
  - tar
  - rsync

Access Control Lists (ACLs)

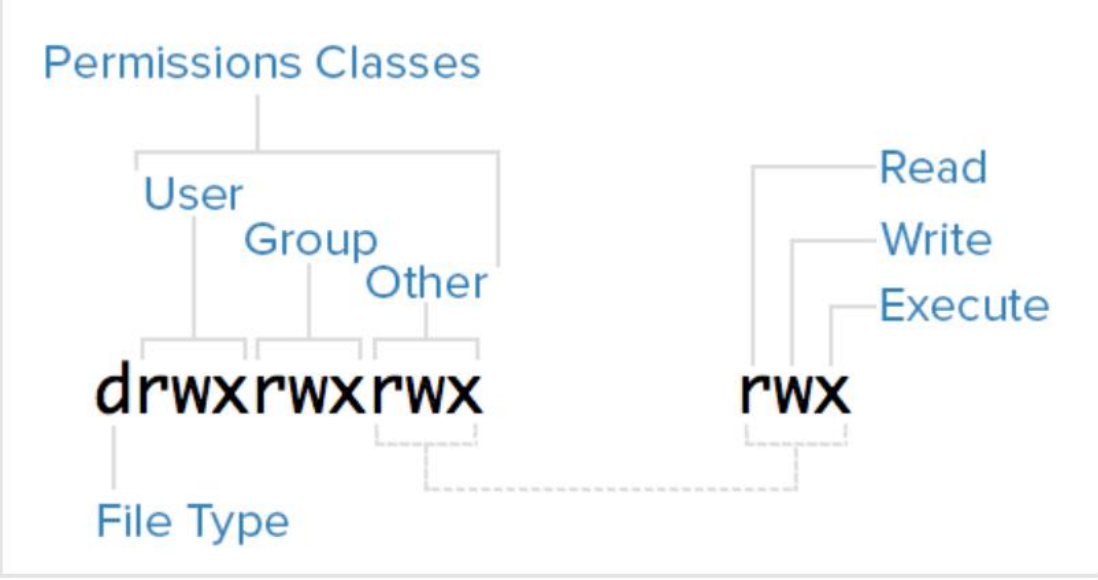
- Access control list (ACL) provides an additional, more flexible permission mechanism for file systems
- ACL allows you to give permissions for any user or group to any disk resource at a granular level
- When to use ACL - think of a scenario in which a particular user is not a member of group but still you want to give the user some read or write access, without making user a member of group
- **setfacl** and **getfacl** are used for setting up ACL and showing ACL respectively

Revision of Linux Permission

- The Linux filesystem gives us three types of permissions:
  1) **U**ser (or user owner)
  2) **G**roup (or group owner)
  3) **O**ther (everyone else)
- With these permissions, we can grant three types of access:
  1) **R**ead
  2) **W**rite
  3) e**X**ecute

Permission


d represents directory
drwxr-xr-x 2 ubuntu ubuntu 80 Sep 6 07:27 Desktop


Permissions Classes
User
Group
Other
Read
Write
Execute
drwxrwxrwx
rwx
File Type

Normal Linux Permissions

```
[student@server home]$ ls -ld salesdoc
drwxrws---. 3 root groupsales 4096 Jul  5 08:05 salesdoc
```

- The salesdoc directory has the following permissions:
  - User owner (root) = read + write + executable (rwx)
  - Group (groupsales) = read + write + executable (rws)
  - Others = no permission (- - -)
- What if we want to give one user who is not in groupsales to have read access?

Using ACL

- To view ACL of file / directory:
  - **`getfacl <filename or directoryname>`**

```
[student@server ~]$ ls -l | grep myusers
-rw-rw-r--. 1 student student   138 Jul  6 19:24 myusers
[student@server ~]$ getfacl myusers
# file: myusers
# owner: student
# group: student
user::rw-
group::rw-
other::r--
```

Setting ACL

1) To add permission for user

```
setfacl -m "u:user:permissions" /path/to/file
```

2) To add permissions for a group

```
setfacl -m "g:group:permissions" /path/to/file
```

3) To allow all files or directories to inherit ACL entries from the directory it is within

```
setfacl -dm "entry" /path/to/dir
```

4) To remove a specific entry

```
setfacl -x "entry" /path/to/file
```

5) To remove all entries

```
setfacl -b path/to/file
```

Using ACL

```
[student@server home]$ ls -ld salesdoc
drwxrws---. 3 root groupsales 4096 Jul  5 08:05 salesdoc
```

```
[student@server ~]$ grep groupsales /etc/group
groupsales:x:672:ali,jim
[student@server ~]$ _
```

- salesdoc directory can be accessed by root and group groupsales
- groupsales has 2 members – ali and jim
- we want to allow user dan to have read+write access to salesdoc by using ACL

Using ACL

To add permission for user dan

`setfacl -m "u:user:permissions" /path/to/file`

```
[root@server ~]# setfacl -m u:dan:rwx /home/salesdoc
[root@server ~]#
[root@server ~]# getfacl /home/salesdoc
getfacl: Removing leading '/' from absolute path names
# file: home/salesdoc
# owner: root
# group: groupsales
# flags: -s-
user::rwx
user:dan:rwx
group::rwx
mask::rwx
other::---
```

Using ACL

- Now dan, who is not a member of groupsales can have access to the salesdoc directory

```
[dan@server ~]$ ls -l /home/salesdoc
total 8
drwxrwsr-x. 2 ali groupsales 4096 Jul  5 08:05 alidir
-rw-rw-r--. 1 ali groupsales   15 Jul  5 07:40 ali.doc
[dan@server ~]$
[dan@server ~]$ touch /home/salesdoc/dan.out
[dan@server ~]$
[dan@server ~]$ ls -l /home/salesdoc
total 8
drwxrwsr-x. 2 ali groupsales 4096 Jul  5 08:05 alidir
-rw-rw-r--. 1 ali groupsales   15 Jul  5 07:40 ali.doc
-rw-rw-r--. 1 dan groupsales    0 Jul  6 21:41 dan.out
[dan@server ~]$ _
```

Set ACL for Group

- Now /home/salesdoc directory can be accessed by groupsales and dan.
- We can also add another group to access /home/salesdoc
- Group groupmktg has 2 members – mike & moses
- We want groupmktg to have Read only access to /home/salesdoc
- Syntax:

```
setfacl -m "g:group:permissions" /path/to/file
```

- Run this command:

```
setfacl -m g:groupmktg:rx  /home/salesdoc
```

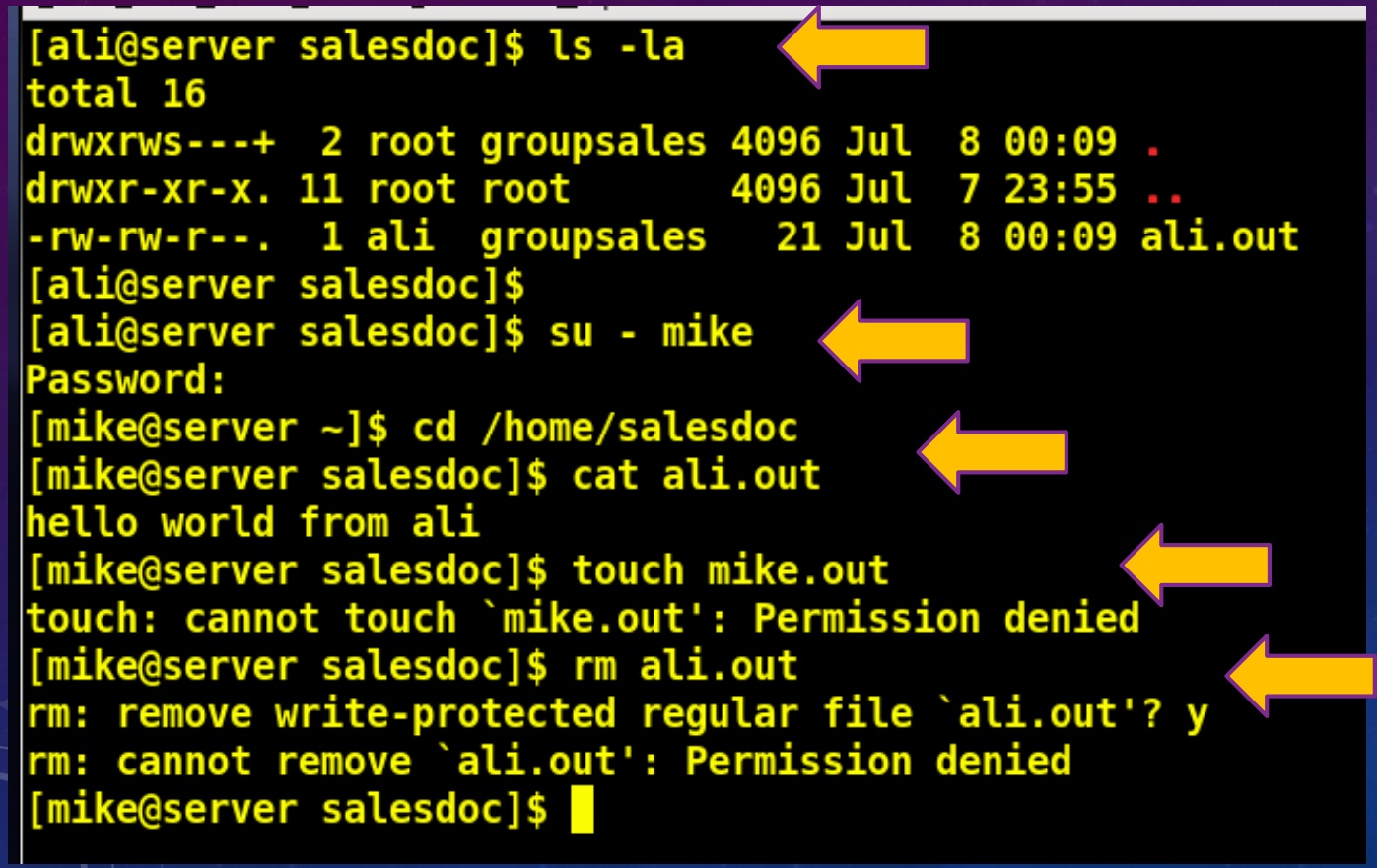
Demo

```
[mike@server salesdoc]$ grep groupsales /etc/group
groupsales:x:506:ali,ben
```

```
[root@server ~]# setfacl -m g:groupmktg:rx /home/salesdoc
[root@server ~]#
[root@server ~]# getfacl /home/salesdoc
getfacl: Removing leading '/' from absolute path names
# file: home/salesdoc
# owner: root
# group: groupsales
# flags: -s-
user::rwx
group::rwx
group:groupmktg:r-x
mask::rwx
other::---
```

```
[mike@server salesdoc]$ grep groupmktg /etc/group
groupmktg:x:507:mike,moses
```

Demo



Mike from groupmktg has only Read access to salesdoc directory

Setup Default ACL

- To set a default ACL, add d: before the rule and specify a directory instead of a file name.
- For example, to set the default ACL for the /share/ directory to read and execute for users not in the user group (an access ACL for an individual file can override it):

```
# setfacl -m d:o:rx /share
```

- ACL mask setting:

https://codingbee.net/rhcsa/rhcsa-the-acls-mask-setting

ACL Status

- You can quickly see if there are ACLs defined on a file (or directory) looking at the **ls –l** output, you will get a “+” sign appearing at the end of the permission columns.

[student@stationX ~]$ **ls –l /shared**

-rw-rwx-r--+ 1 student student 0 2009-06-25 00:52 schedular.txt

```
[mike@server salesdoc]$ ls -l /home | grep salesdoc
drwxrws---+  2 root    groupsales 4096 Jul  8 00:09 salesdoc
[mike@server salesdoc]$
```

Backup using tar

- Can use tar command to archive files to a device, such as a hard drive or tape.
- The tar program creates an archive file that can contain other directories and files and (optionally) compress the archive for efficient storage.
- Syntax:

  tar [*options*] <*destination*> <*source*>

  Example:

  tar -cvpzf /tmp/backup.tar.gz /home

Backup using tar

- Backup the /home directory and all its sub-directories to /tmp/backup.tar.gz file

```
tar -cvpzf /tmp/backup.tar.gz /home
```

where

c = create a new archive

v = verbose

p = preserve the permissions

z = compress the tar archive using gzip

f = specify name of archive file

List the Archive Files

- List the archive file /tmp/backup.tar.gz

```
tar -tzf /tmp/backup.tar.gz
```

where

t = list the contents of the archive

z = uncompress the tar archive using gzip

f = specify name of archive file

Recover Files using tar

- Extract the archive file /tmp/backup.tar.gz to another location

```
tar -xvzf /tmp/backup.tar.gz -C /tmp/recover
```

where

x = extract files from the archive

z = uncompress the tar archive using gzip

f = specify name of archive file

v = verbose

Note: /tmp/recover must exist

Backup using rsync

- Rsync is a file coping tool that can copy files between a local system and remote system
- Incremental backup - rsync only needs to copy the differences between the systems
- Syntax:

```
rsync [options] <source>/ <destination>
```

- If destination does not exist, rsync will create it (compare with tar)

rsync Demo

- Backup home directory to /tmp/homebackup

```
rsync -avz /home/student/ /tmp/studentbackup
```

where:

- a = archive mode; equals -rlptgoD (no -H,-A,-X)
- v = verbose
- z = compress mode during transmission
- Note: / at the end of the source. This is to tell rsync that the source is all the contents of the source directory. If the “/” at the end of the source directory is missing, rsync will simply create a copy of the source directory instead of its contents.
- If /tmp/studentbackup does not exist, it will be created
- You must have the necessary permissions to copy

rsync Incremental Backup

- rsync will copy only changes after a first backup

Example:

- Executed **`rsync -avz /home/student/ /tmp/student1`**
- Make changes – create new directory newdir and modified file
- If run rsync again, only changes will be backup

```
[student@server ~]$ rsync -avz /home/student/ /tmp/student1
sending incremental file list
./
file
newdir/

sent 10360 bytes  received 189 bytes  21098.00 bytes/sec
total size is 19195566  speedup is 1819.66
```

Demo Remote rsync

- Backup home directory at Client to Server /tmp

```
[student@client Desktop]$ rsync -avze ssh /home/student/ server:/tmp/zion
student@server's password:
```

```
sent 1189505 bytes  received 3785 bytes  159105.33 bytes/sec
total size is 19420107  speedup is 16.27
```

```
[student@server ~]$ ls -l /tmp/zion
total 4
drwx------. 31 student student 4096 Jul  8 20:46 student
```

Summary

- Macro permission -
  - user/owner, group, other
  - read, write and execute
- Micro permission – ACL
- Backup and Recovery
  - tar
  - Rsync
  - others