12-06-2020
https://youtu.be/dnuAu24KUuE

Permissions

File permission

[root@zmpt01 ~]# ls -l
total 0
-rw-r--r--. 1 root root 0 Dec 6 13:22 file

-rw-r--r--. 1 root root 0 Dec 6 13:22 file

Ver first '-' is not part of permissions, but indicates it's a file1

User – u			Group – g			Other – o		
Read	Write	Execute	Read	Write	Execute	Read	Write	Execute
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

Read = 4
Write = 2
Execute = 1

Read = read the file, read permissions
Write = write into the file, edit permissions
Execute = for script file, execute file, deleting file or folder

Number value of permission for file1

-rw-r--r--. 1 root root 0 Dec 6 13:22 file

User	Group	Others
rw-	r- -	r- -
6	4	4

So the final permission number for file1 is 644

644 is also default permission for file in Linux System

Modify the permissions

Grant ‘rw’ permission for all

Number for read write is 6 = 4 + 2

rw- = 6
rw- = 6
rw- = 6

```
[root@zmpt01 ~]# chmod 666 file1          #< ---chmod will modify the permissions
[root@zmpt01 ~]# ls -l
total 0
-rw-rw-rw-. 1 root root 0 Dec  6 13:34 file1
```

Grant additional execute permission to group

rw-rwxrw-

rw- = 6
rwx = 7
rw- = 6

```
[root@zmpt01 ~]# chmod 676 file1
[root@zmpt01 ~]# ls -l
total 0
-rw-rwxrw-. 1 root root 0 Dec  6 13:34 file1
```

Take away all the permissions from everybody

--------- = 000

```
[root@zmpt01 ~]# chmod 000 file1
[root@zmpt01 ~]# ls -l
total 0
----------. 1 root root 0 Dec  6 13:34 file1
```

regardless of permissions set, root user has unrestricted permisions

Note: if you are modifying permissions using number, it must be three digits

Changing permissions using associated letter

Grant read and write permission only User/ owner

-rw------- = 600 = u+rw

```
[root@zmpt01 ~]# chmod u+rw file1
[root@zmpt01 ~]# ls -l
total 0
-rw-------. 1 root root 0 Dec  6 13:34 file1
```

Grant everybody read write permissions

-rw-rw-rw- = 666 = ugo+rw

```
[root@zmpt01 ~]# chmod ugo+rw file1
[root@zmpt01 ~]# ls -l
total 0
-rw-rw-rw-. 1 root root 0 Dec  6 13:34 file1
```

-rw-rw-rw-

Remove write permission for others

-rw-rw-r-- = 664 = o-w

```
[root@zmpt01 ~]# chmod o-w file1
[root@zmpt01 ~]# ls -l
total 0
-rw-rw-r--. 1 root root 0 Dec  6 13:34 file1
```

Full permissions

```
[root@zmpt01 ~]# chmod ugo+rwx file1
[root@zmpt01 ~]# ls -l
total 0
```

-rwxrwxrwx. 1 root root 0 Dec 6 13:34 file1

Informational

[root@zmpt01 ~]# ls -l
total 0
d---------. 2 root root 6 Dec 6 14:09 dir1
-rwxrwxrwx. 1 root root 0 Dec 6 13:34 file1

Look at the permissions you can only see the user and groups, other are not listed

User – Yellow
Group - Orange

Directory Permissions

drwxr-xr-x. 2 root root 6 Dec 6 14:09 dir1

very first 'd' is not part of permission – it indicates directory

User – u			Group – g			Other – o		
Read	Write	Execute	Read	Write	Execute	Read	Write	Execute
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

Read = 4
Write = 2
Execute = 1

Read = read the files in the directory
Write = creating new files in the directory
Execute = going inside the directory

Number value of permission for file1

drwxr-xr-x. 2 root root 6 Dec 6 14:09 dir1

User	Group	Others
rwx	r- x	r- x
7	5	5

So the final permission number for dir1 is 755

755 is also default permission for file in Linux System

[root@zmpt01 ~]# mkdir /userdir

Remove all access to directory

drwxr-xr-x. 2 root root 6 Dec 6 14:25 /userdir

[root@zmpt01 ~]# chmod 000 /userdir
[root@zmpt01 ~]# ls -ld /userdir/
d---------. 2 root root 6 Dec 6 14:25 /userdir/

regardless of permissions set, root user has unrestricted permisions

[terminator@zmpt01 ~]$ cd /userdir/
-bash: cd: /userdir/: Permission denied

--------x = 001 = o+x

Grant execute permissions only to others

[root@zmpt01 ~]# chmod o+x /userdir/
[root@zmpt01 ~]# ls -ld /userdir/
d--------x. 2 root root 6 Dec 6 14:25 /userdir/

[terminator@zmpt01 ~]$ cd /userdir/
[terminator@zmpt01 userdir]$ pwd
/userdir

UMASK

total 0
drwxr-xr-x. 2 root root 6 Dec 6 14:59 dir2 #< ---755 is the default Directory permission
-rw-r--r--. 1 root root 0 Dec 6 14:59 file2 #< ---644 is the default File permission

This is set by default umask

By default system provides 644 permissions to file
By default system provides 755 permissions to directory

```
[root@zmpt01 ~]# umask
0022
```

Symbolic	Users	Group	Others
0	0	2	2

File permission

File permission are based on 666

	Users	Group	Others	Default
System provided permissions	6	6	6	666
Umask – removes the permission	0	2	2	022
Final defualt permissions	6	4	4	644

Directory permission

Directory permission are on 777

	Users	Group	Others	Default
System provided permissions	7	7	7	777
Umask – removes the permission	0	2	2	022
Final defualt permissions	7	5	5	755

Lets set umask to 0000

```
[root@zmpt01 ~]# umask 0000
[root@zmpt01 ~]# umask
0000
```

Note: system goes back to default umask when the system reboots

```
[root@zmpt01 ~]# touch file3
```

[root@zmpt01 ~]# ls –l
-rw-rw-rw-. 1 root root 0 Dec 6 15:11 file3

Permission is 666

[root@zmpt01 ~]# mkdir dir3
[root@zmpt01 ~]# ls –l
drwxrwxrwx. 2 root root 6 Dec 6 15:12 dir3

permission is 777

Group permissions

Linux groups is a mechanism to manage a large collection of users and mange their permissions. All linux users have a **User ID** (UID) as well as **Group ID** (GID) by default

[root@zmpt01 ~]# id terminator
uid=1000(terminator) gid=1000(terminator) groups=1000(terminator),10(wheel)

UID – User Id	GID – user primary group	Group User is part of
uid=1000(terminator)	gid=1000(terminator)	groups=1000(terminator), 10(wheel)

Create new group

[root@zmpt01 ~]# groupadd -g 9000 machine

Command	Primary group	GID	Group Name
groupadd	-g	9000	machine

Add user to the group

[root@zmpt01 ~]# usermod -aG machine terminator

Command	-a Add -G secondary groug	GID/ name	UID/ name
usermod	-aG	Machine	terminator

[root@zmpt01 ~]# usermod -g 9000 terminator < ---Set primary group using -g
[root@zmpt01 ~]# id terminator
uid=1000(terminator) gid=9000(machine) groups=9000(machine),10(wheel)

[root@zmpt01 ~]# id terminator
uid=1000(terminator) gid=1000(terminator) groups=1000(terminator),10(wheel),9000(machine)

UID – User Id	GID – user primary group	Group User is part of
uid=1000(terminator)	gid=1000(terminator)	groups=1000(terminator), 10(wheel) 9000(machine)

Assign folder permisions

Create folder

[root@zmpt01 ~]# mkdir /assignment
[root@zmpt01 ~]# ls -ld /assignment/
drwxr-xr-x. 2 root root 6 Dec 6 15:31 /assignment/

Terminator is unable to create file1 in /assignment

[terminator@zmpt01 assignment]$ touch file1
touch: cannot touch 'file1': Permission denied

Change the group ownership to machine

[root@zmpt01 ~]# chgrp 9000 /assignment/
[root@zmpt01 ~]# ls -ld /assignment/
drwxr-xr-x. 2 root machine 6 Dec 6 15:31 /assignment/ #< ---Machine is group owner of /assignment

Command	GID	Folder
Chgrp	9000	/assignment

[root@zmpt01 ~]# chown :machine /assignment/ #< --- another way of changing group owner of /assignment
[root@zmpt01 ~]# ls -ld /assignment/

Change the group permissions

```
[root@zmpt01 ~]# chmod 775 /assignment/
[root@zmpt01 ~]# ls -ld /assignment/
drwxrwxr-x. 2 root machine 6 Dec  6 15:31 /assignment/      #< ---Machine group has rwx permissions
```

```
[terminator@zmpt01 assignment]$ touch file1
[terminator@zmpt01 assignment]$ ls -l
total 0
-rw-rw-r--. 1 terminator terminator 0 Dec  6 15:38 file1
```

Remove user from the group

```
[root@zmpt01 ~]# gpasswd -d terminator wheel
Removing user terminator from group wheel
```

Command	Delete	User id	Group id
Gpasswd	-d	Terminator	Wheel

```
[root@zmpt01 ~]# id terminator
uid=1000(terminator) gid=9000(machine) groups=9000(machine)
```

```
[terminator@zmpt01 ~]$ sudo su -
[sudo] password for terminator:
terminator is not in the sudoers file.  This incident will be reported.
```

12-12-2020
https://youtu.be/gCipZh-NsSs

Set Group ID

SGID: all the files that are created in a directory with SGID set belongs to the group to with the directory belongs. Not the group user belong.

- Special permission set for directories
- When ever the user creates a file and direcories insde the SGID configure folder, it will inherit the Group Ownership of the SGID directory
- The group ownership is not rectro active.

```
[root@zmpt01 ~]# ls -ld /DATA/
```

drwxr-xrwx. 2 root humans 6 Dec 12 14:21 /DATA/

[terminator@zmpt01 DATA]$ ls -l
total 0
-rw-r--r--. 1 terminator machine 0 Dec 12 14:22 file1

[root@zmpt01 ~]# chmod g+s /DATA/ #< --- set the SGID

[root@zmpt01 ~]# ls -ld /DATA/
drwxr-srwx. 2 root humans 19 Dec 12 14:22 /DATA/ #< --- 's' is indication the SGID is on folder /DATA

[terminator@zmpt01 DATA]$ touch file2
[terminator@zmpt01 DATA]$ ls –l
-rw-r--r--. 1 terminator humans 0 Dec 12 14:26 file2

[spiderman@zmpt01 DATA]$ ls –l
-rw-rw-r--. 1 spiderman humans 0 Dec 12 14:38 file4

[root@zmpt01 ~]# chmod g-x /DATA/

[root@zmpt01 ~]# ls -ld /DATA/
drwxr-Srwx. 2 root humans 97 Dec 12 15:07 /DATA/ #< --- 'S' upper case S without group execute permission

Set UID - SUID

SUID: the command inherits the owners execute permissions

- SUID is used for the files for the execution purpose
- It inherits the owners/root execute permission
- Regular user does not have permissions
- But still can update the file when executing the command

[root@zmpt01 ~]# ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 27856 Aug 8 2019 /usr/bin/passwd #< --Typical example for user running with root permission

Generally passwd command is allowed for regular user, but this command is editing /etc/shadow file

[root@zmpt01 ~]# ls -l /etc/shadow
----------. 1 root root 979 Dec 12 15:28 /etc/shadow #< ---Looking at the permission, no one has write permissions

[terminator@zmpt01 ~]$ passwd
Changing password for user terminator.
Changing password for terminator.
(current) UNIX password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully.

[root@zmpt01 ~]# ls -l /etc/shadow
----------. 1 root root 979 Dec 12 15:43 /etc/shadow #< --- file got update even though Terminator does not have any perm

[root@zmpt01 ~]# ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 27856 Aug 8 2019 /usr/bin/passwd
[root@zmpt01 ~]# chmod u-s /usr/bin/passwd
[root@zmpt01 ~]# ls -l /usr/bin/passwd
-rwxr-xr-x. 1 root root 27856 Aug 8 2019 /usr/bin/passwd

User is unable to change the password

[terminator@zmpt01 ~]$ passwd
Changing password for user terminator.
Changing password for terminator.
(current) UNIX password:
New password:
Retype new password:
passwd: Authentication token manipulation error

[root@zmpt01 ~]# chmod u+s /usr/bin/passwd

User is able change the password now

[terminator@zmpt01 ~]$ passwd

Changing password for user terminator.
Changing password for terminator.
(current) UNIX password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully.

[root@zmpt01 ~]# ls -l /usr/bin/sudo
---s--x--x. 1 root root 147320 Aug 8 2019 /usr/bin/sudo

[root@zmpt01 ~]# chmod u-s /usr/bin/sudo
[root@zmpt01 ~]# ls -l /usr/bin/sudo
---x--x--x. 1 root root 147320 Aug 8 2019 /usr/bin/sudo

[root@zmpt01 ~]# id terminator
uid=1000(terminator) gid=9000(machine) groups=9000(machine)
[root@zmpt01 ~]# usermod -aG wheel terminator

[root@zmpt01 ~]# id terminator
uid=1000(terminator) gid=9000(machine) groups=9000(machine),10(wheel)

[terminator@zmpt01 ~]$ sudo su -
sudo: /usr/bin/sudo must be owned by uid 0 and have the setuid bit set

[root@zmpt01 ~]# chmod u+s /usr/bin/sudo

[terminator@zmpt01 ~]$ sudo su -
[sudo] password for terminator:
Last login: Sat Dec 12 16:05:56 EST 2020 on pts/1

12-13-2020
https://youtu.be/A7uNJFZhTro

Sticky Bit

Sticky Bit – it is a delete protection, if you are not root or owner of the file you cannot delete a file. This is folder level permission

```
[root@zmpt01 ~]# ls -ld /tmp
drwxrwxrwt. 8 root root 172 Dec 13 03:19 /tmp
```

*by default **/tmp** folder is set with stick bit permissions*

```
[root@zmpt01 ~]# mkdir /DATA1
[root@zmpt01 ~]# chmod 777 /DATA1/
```

```
[terminator@zmpt01 DATA1]$ touch file1
[terminator@zmpt01 DATA1]$ ls -l
total 0
-rw-r--r--. 1 terminator machine 0 Dec 13 13:45 file1
```

```
[spiderman@zmpt01 DATA1]$ rm file1
rm: remove write-protected regular empty file ‘file1’? y   #< ---Spiderman successfully detelet the file
[spiderman@zmpt01 DATA1]$ ls -l
total 0
```

Assign sticky bit to /DATA1

```
[root@zmpt01 ~]# chmod o+t /DATA1/
[root@zmpt01 ~]# ls -ld /DATA1/
drwxrwxrwt. 2 root root 6 Dec 13 13:45 /DATA1/   < -- now the folder protech with sticky bit
```

```
[terminator@zmpt01 DATA1]$ touch file1
[terminator@zmpt01 DATA1]$ touch file2
[terminator@zmpt01 DATA1]$ touch file3
```

```
[spiderman@zmpt01 DATA1]$ rm file1
rm: remove write-protected regular empty file ‘file1’? y
rm: cannot remove ‘file1’: Operation not permitted
```

FACL – File Access Control List

```
[root@zmpt01 ~]# mkdir /BANK
[root@zmpt01 ~]# ls -ld /BANK/
drwxr-xr-x. 2 root root 6 Dec 13 13:54 /BANK/
```

```
[root@zmpt01 ~]# getfacl /BANK/
getfacl: Removing leading '/' from absolute path names
# file: BANK/
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

```
[terminator@zmpt01 DATA1]$ cd /BANK/
[terminator@zmpt01 BANK]$ ls -l
total 0
[terminator@zmpt01 BANK]$ touch file1
touch: cannot touch ‘file1’: Permission denied          #< --- expected denial for other users
```

Grant permission to specific user on folder

```
[root@zmpt01 ~]# setfacl -m u:terminator:rwx /BANK/
[root@zmpt01 ~]# getfacl /BANK/
getfacl: Removing leading '/' from absolute path names
# file: BANK/
# owner: root
# group: root
user::rwx
user:terminator:rwx
group::r-x
mask::rwx
other::r-x

[root@zmpt01 ~]# ls -ld /BANK/
drwxrwxr-x+ 2 root root 19 Dec 13 14:00 /BANK/        #< --- + is indication of FACL
```

```
[terminator@zmpt01 BANK]$ touch file1
[terminator@zmpt01 BANK]$ ls -l
total 0
-rw-r--r--. 1 terminator machine 0 Dec 13 14:00 file1
```

Grant permission on individual file

```
[terminator@zmpt01 BANK]$ setfacl -m u:spiderman:rwx file1
```

```
[terminator@zmpt01 BANK]$ getfacl file1
# file: file1
# owner: terminator
# group: machine
user::rw-
user:spiderman:rwx
group::r--
mask::rwx
other::r--

[terminator@zmpt01 BANK]$ ls -l
total 0
-rw-rwxr--+ 1 terminator machine 0 Dec 13 14:00 file1
```

```
[spiderman@zmpt01 BANK]$ cat file1
this is file1 content
this is file1 content
this is file1 content

[spiderman@zmpt01 BANK]$ vi file1      #< ---user spiderman is able read and write to the file
```

Permissions

Identity	
User	u
Group	g
Other	o
All	a

Permission		
Read	r	4
Write	w	2
Execute	x	1

Actions	
+	Add permission
'-'	Remove permission
=	Make it only permission

Examples

Permission	Information

g+w	adds write access for the group
o-rwx	removes all permissions for others
u+x	allows the file owner to execute the file
a+rw	allows everyone to read and write to the file
ug+r	allows the owner and group to read the file
g=rx	allows only the group to read and execute (not write
g+w	adds write access for the group
g=rx	allows only the group to read and execute (not write)

Permission	Numerical	Information
-rw-------	600	Only the owner has read and write permissions.
-rw-r--r--	644	Only the owner has read and write permissions; the group and others have read only. DEFAULT
-rwx------	700	Only the owner has read, write, and execute permissions.
-rwxr-xr-x	755	The owner has read, write, and execute permissions; the group and others have only read and execute.
-rwx--x--x	711	The owner has read, write, and execute permissions; the group and others have only execute.
-rw-rw-rw-	666	Everyone can read and write to the file. (Be careful with these permissions.)
-rwxrwxrwx	777	Everyone can read, write, and execute. (Again, this permissions setting can be hazardous.)

Chmod

```
[root@zmpt01 ~]# ls -l
total 0
----------. 1 root root 0 Oct 11 12:17 file1
[root@zmpt01 ~]# chmod o+w file1
[root@zmpt01 ~]# ls -l
total 0
--------w-. 1 root root 0 Oct 11 12:17 file1

[root@zmpt01 ~]# chmod a+r file1
[root@zmpt01 ~]# ls -l
total 0
-r--r--rw-. 1 root root 0 Oct 11 12:17 file1
```

```
[root@zmpt01 ~]# chmod a+rwx file1
[root@zmpt01 ~]# ls -l
total 0
-rwxrwxrwx. 1 root root 0 Oct 11 12:17 file1

[root@zmpt01 ~]# chmod a-x file1
[root@zmpt01 ~]# ls -l
total 0
-rw-rw-rw-. 1 root root 0 Oct 11 12:17 file1

[root@zmpt01 ~]# chmod a-rwx file1
[root@zmpt01 ~]# ls -l
total 0
----------. 1 root root 0 Oct 11 12:17 file1
```

```
[root@zmpt01 ~]# chmod 002 file1
[root@zmpt01 ~]# ls -l
total 0
--------w-. 1 root root 0 Oct 11 12:17 file1

[root@zmpt01 ~]# chmod 446 file1
[root@zmpt01 ~]# ls -l
total 0
-r--r--rw-. 1 root root 0 Oct 11 12:17 file1

[root@zmpt01 ~]# chmod 777 file1
[root@zmpt01 ~]# ls -l
total 0
-rwxrwxrwx. 1 root root 0 Oct 11 12:17 file1

[root@zmpt01 ~]# chmod 666 file1
[root@zmpt01 ~]# ls -l
total 0
-rw-rw-rw-. 1 root root 0 Oct 11 12:17 file1

[root@zmpt01 ~]# chmod 000 file1
[root@zmpt01 ~]# ls -l
total 0
----------. 1 root root 0 Oct 11 12:17 file1
```

Chage

```
[root@zmpt01 ~]# chage -l terminator
Last password change                    : Dec 12, 2020
```

Password expires : never
Password inactive : never
Account expires : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7

Last password change : Dec 12, 2020	Date of the password was changed
Password expires : never	Password expirartion date
Password inactive : never	Password inactive date
Account expires : never	Id expiration date
Minimum number of days between password change : 0	When will next password be force to change
Maximum number of days between password change : 99999	Maximum days between the next password change
Number of days of warning before password expires : 7	Warning period

Default setting for the pasword age

PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_MIN_LEN 5
PASS_WARN_AGE 7

[root@zmpt01 ~]# vi /etc/login.defs

PASS_MAX_DAYS 60
PASS_MIN_DAYS 0
PASS_MIN_LEN 5
PASS_WARN_AGE 7

[root@zmpt01 ~]# useradd ironman
[root@zmpt01 ~]# chage -l ironman
Last password change : Dec 13, 2020
Password expires : Feb 11, 2021
Password inactive : never
Account expires : never
Minimum number of days between password change : 0
Maximum number of days between password change : 60
Number of days of warning before password expires : 7

Set passwordl to never expire

```
[root@zmpt01 ~]# chage -m 0 -M 99999 -I -1 -E -1 ironman



[root@zmpt01 ~]# chage -l ironman
Last password change                                    : Dec 13, 2020
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

Chown- change ownership

Chown is to change/assign ownership of the files and folders to users and groups

```
[root@zmpt01 ~]# mkdir /DATA2
drwxr-xr-x. 2 root root 6 Dec 13 15:14 /DATA2/
```

Change the file ownership

```
[root@zmpt01 DATA2]# touch file1

[root@zmpt01 DATA2]# ls -l file1
-rw-r--r--. 1 root root 0 Dec 13 15:15 file1

[root@zmpt01 DATA2]# chown terminator file1
[root@zmpt01 DATA2]# ls -l file1
-rw-r--r--. 1 terminator root 0 Dec 13 15:15 file1
```

Change the gorup ownership

```
[root@zmpt01 DATA2]# chown :machine file1
[root@zmpt01 DATA2]# ls -l file1
-rw-r--r--. 1 terminator machine 0 Dec 13 15:15 file1
```

Change user and group ownership

```
[root@zmpt01 DATA2]# chown spiderman:superhero file1
[root@zmpt01 DATA2]# ls -l file1
```

-rw-r--r--. 1 spiderman superhero 0 Dec 13 15:15 file1

Change the user ownership of Folder

[root@zmpt01 ~]# chown spiderman /DATA2/
[root@zmpt01 ~]# ls -ld /DATA2/
drwxr-xr-x. 2 spiderman root 19 Dec 13 15:15 /DATA2/

Chgrp – change group

Chgrp – allow to change group only

[root@zmpt01 DATA2]# ls -l file1
-rw-r--r--. 1 spiderman machine 0 Dec 13 15:15 file1

Change group ownerhsip of folder

[root@zmpt01 ~]# chgrp machine /DATA2/

[root@zmpt01 ~]# ls -ld /DATA2/
drwxr-xr-x. 2 spiderman machine 19 Dec 13 15:15 /DATA2/