UNDERSTANDING LINUX FILE PERMISSION WITH EXPERT USING REAL WORLD EXAMPLES.

Linux file permissions are the **first line of defense** in any system. Understanding them is critical because most servers and security tools run on Linux.

If you don't understand file permissions:

❌ You cannot secure a server

❌ You cannot investigate attacks

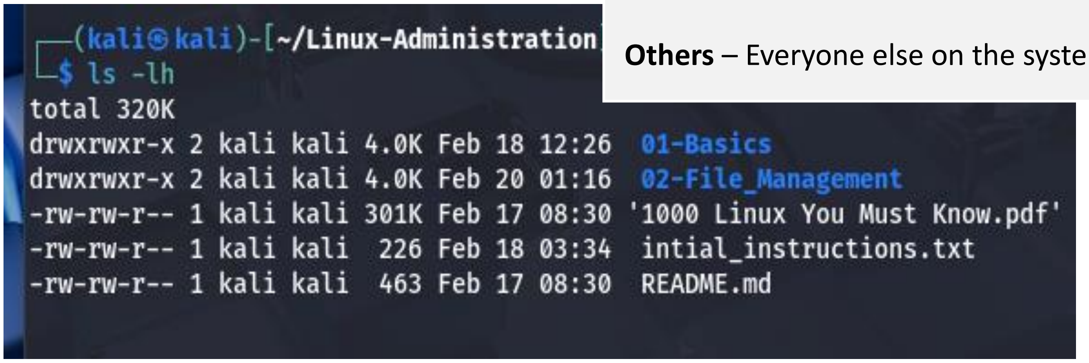
❌ You cannot fix privilege problems

Every file in Linux has:

Owner (User) – The person who created the file.

Group – A set of users who share access.

Others – Everyone else on the system.



```
┌──(kali㉿kali)-[~/Linux-Administration]
└─$ ls -lh
total 320K
drwxrwxr-x 2 kali kali 4.0K Feb 18 12:26  01-Basics
drwxrwxr-x 2 kali kali 4.0K Feb 20 01:16  02-File_Management
-rw-rw-r-- 1 kali kali 301K Feb 17 08:30 '1000 Linux You Must Know.pdf'
-rw-rw-r-- 1 kali kali  226 Feb 18 03:34  intial_instructions.txt
-rw-rw-r-- 1 kali kali  463 Feb 17 08:30  README.md
```

Above is the result of ls -lh command in CLI, let use row four

```
-rw-rw-r--  1  kali  kali  301K  Feb 17 08:30  '1000 Linux You Must Know.pdf'
↓           ↓  ↓     ↓     ↓     ↓             ↓
FileType+  Links Owner Group Size  Date & Time  File Name
Permissions
```

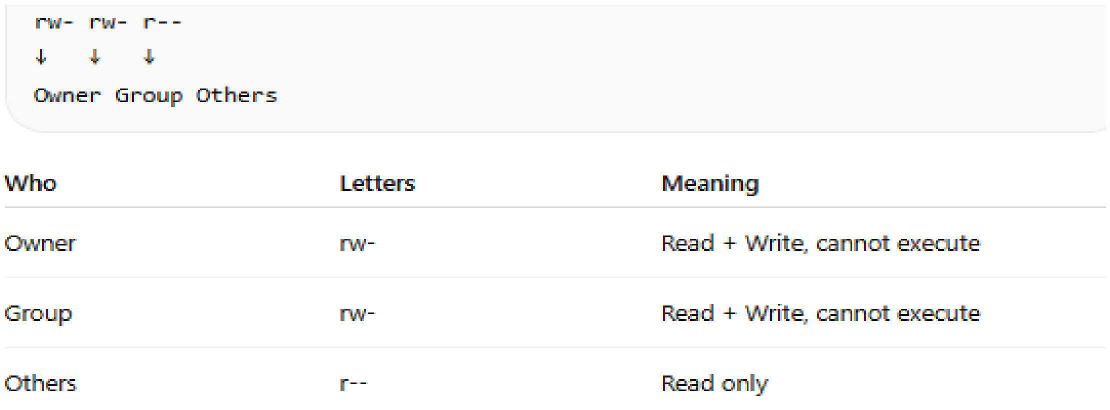
File type:

First character (-) → File type:

- - = regular file
- d = directory
- l = symbolic link

From this you can know if the file is a folder, regular file like: Text, Binary, Images, Programs / scripts or directory or link

FILE PERMISSION

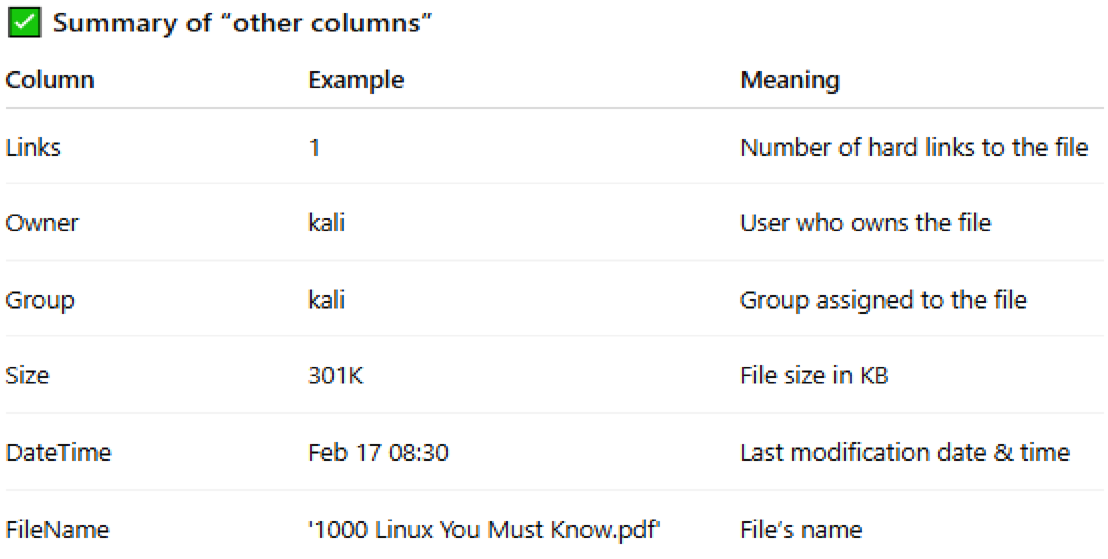


```
rw- rw- r--
↓   ↓   ↓
Owner Group Others
```

Who	Letters	Meaning
Owner	rw-	Read + Write, cannot execute
Group	rw-	Read + Write, cannot execute
Others	r--	Read only

In permission screen short it clear that all user cannot execute, owner can read and write as well as group but others can only read. Very key for Access control in real world servers.

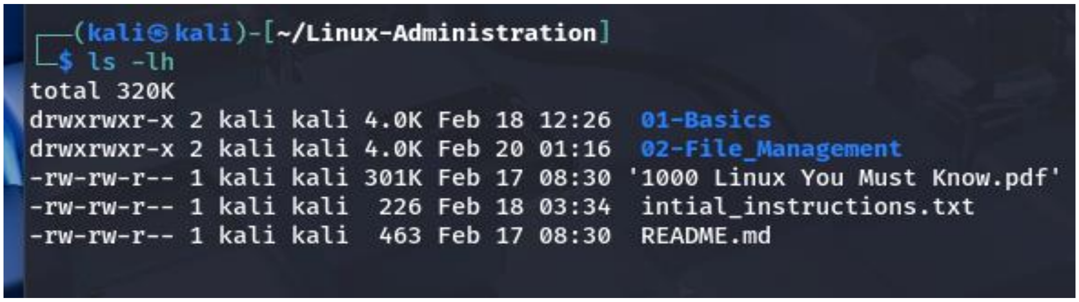
We will dive late on changing permissions, let procced to next column in summary



✅ Summary of "other columns"

Column	Example	Meaning
Links	1	Number of hard links to the file
Owner	kali	User who owns the file
Group	kali	Group assigned to the file
Size	301K	File size in KB
DateTime	Feb 17 08:30	Last modification date & time
FileName	'1000 Linux You Must Know.pdf'	File's name

Hope with above each column is understood try to name is column:



```
┌──(kali㉿kali)-[~/Linux-Administration]
└─$ ls -lh
total 320K
drwxrwxr-x 2 kali kali 4.0K Feb 18 12:26  01-Basics
drwxrwxr-x 2 kali kali 4.0K Feb 20 01:16  02-File_Management
-rw-rw-r-- 1 kali kali 301K Feb 17 08:30 '1000 Linux You Must Know.pdf'
-rw-rw-r-- 1 kali kali  226 Feb 18 03:34  intial_instructions.txt
-rw-rw-r-- 1 kali kali  463 Feb 17 08:30  README.md
```

DEEP DIVE INTO FILE PERMISSIONS AND PERMISSION MANAGEMENT

Using same example

```
-rw-rw-r-- 1 kali kali 301K Feb 17 08:30 '1000 Linux You Must Know.pdf'
```

We can understand permission management in two ways:

1. Binary/ Numeric Methods
2. Symbolic/Human-Readable

1. Binary/Numeric method

Permissions are written rwx (read, write, execute) converting lines to binary

r-4

w-2

x-1

-no permission

You **add the numbers** for each group:

- rwx → 4 + 2 + 1 = 7
- rw- → 4 + 2 + 0 = 6
- r-- → 4 + 0 + 0 = 4
- --x → 0 + 0 + 1 = 1

We have three groups owner, group and other so changing permission we will user chmod

Example

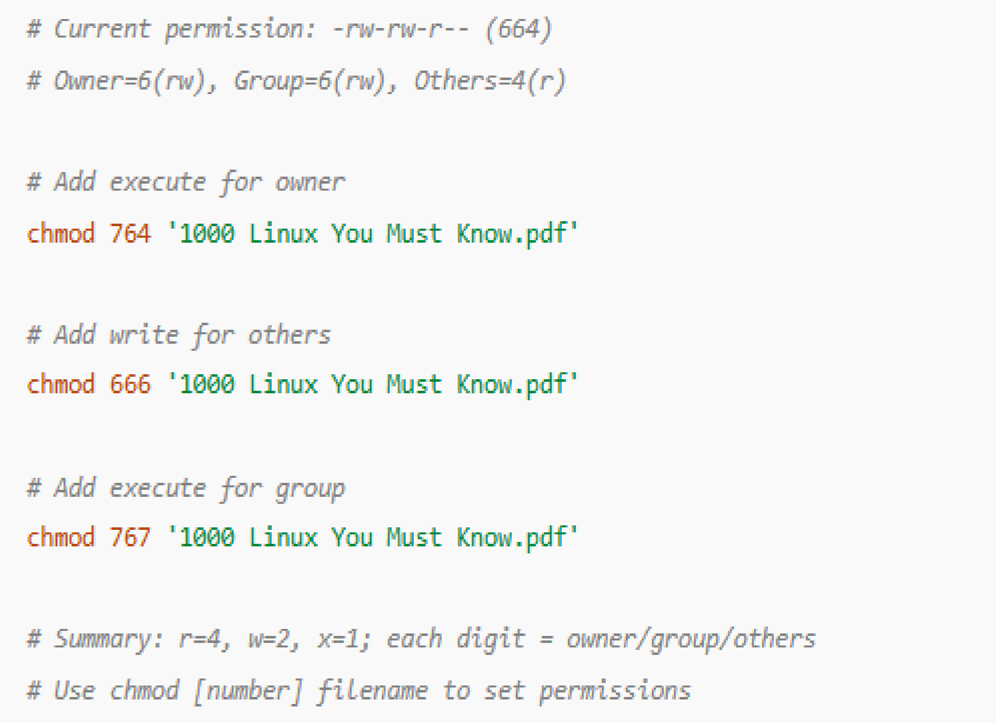
```
-rw-rw-r-- 1 kali kali 301K Feb 17 08:30 '1000 Linux You Must Know.pdf'
```

Permission per group is 661 to add or deny permission use chmod command

Syntax

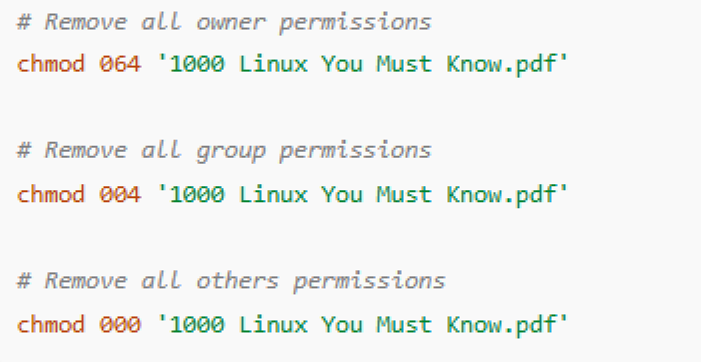
```
chmod [permissions] [filename]
```

Example of addicting permissions



```
# Current permission: -rw-rw-r-- (664)
# Owner=6(rw), Group=6(rw), Others=4(r)

# Add execute for owner
chmod 764 '1000 Linux You Must Know.pdf'

# Add write for others
chmod 666 '1000 Linux You Must Know.pdf'

# Add execute for group
chmod 767 '1000 Linux You Must Know.pdf'

# Summary: r=4, w=2, x=1; each digit = owner/group/others
# Use chmod [number] filename to set permissions
```

Denying or Removing



```
# Remove all owner permissions
chmod 064 '1000 Linux You Must Know.pdf'

# Remove all group permissions
chmod 004 '1000 Linux You Must Know.pdf'

# Remove all others permissions
chmod 000 '1000 Linux You Must Know.pdf'
```

With example you can now say permission for each group below

1. chmod 777 file.pdf …………………………………………………. Ans rwx rwx rwx
2. chmod 646 file.pdf…………………………………………………. Ans r-x r—r-x
3. chmod 576 file.pdf ………………………………………………….

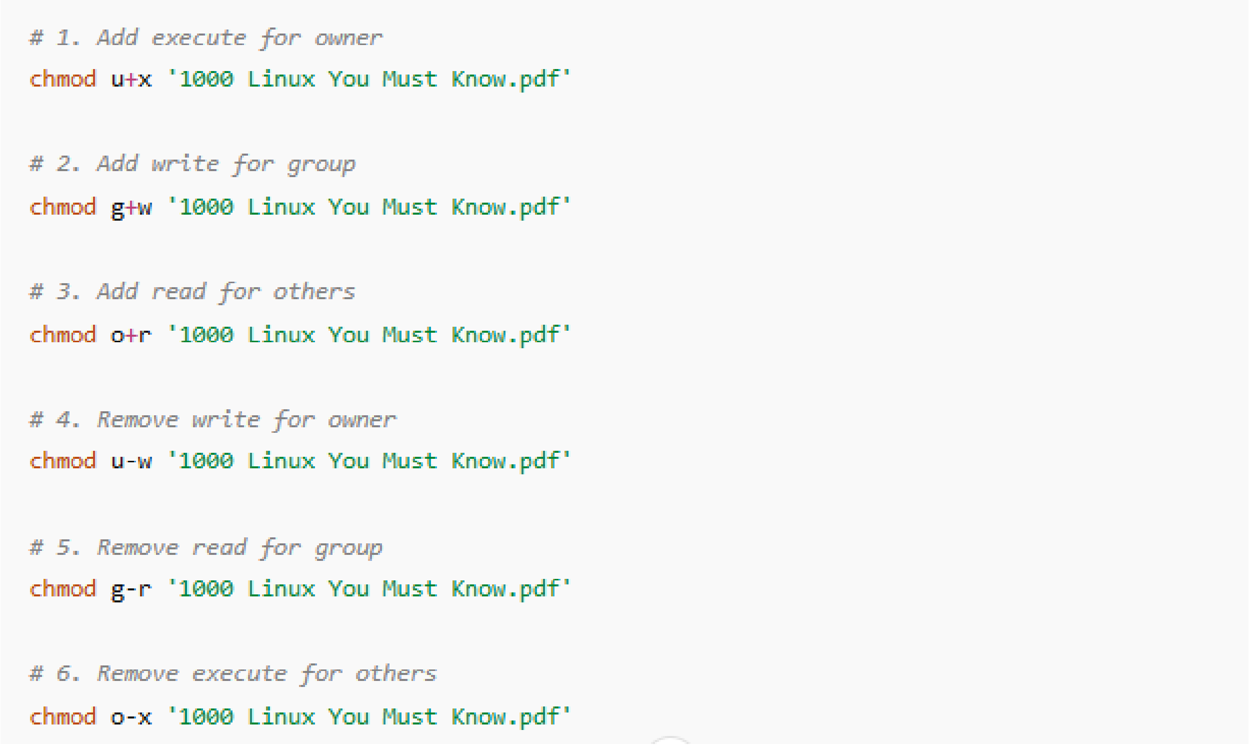
4. chmod 427 file.pdf…………………………………………………...
5. chmod 007 file.pdf …………………………………………………

3. Symbolic/Human-Readable

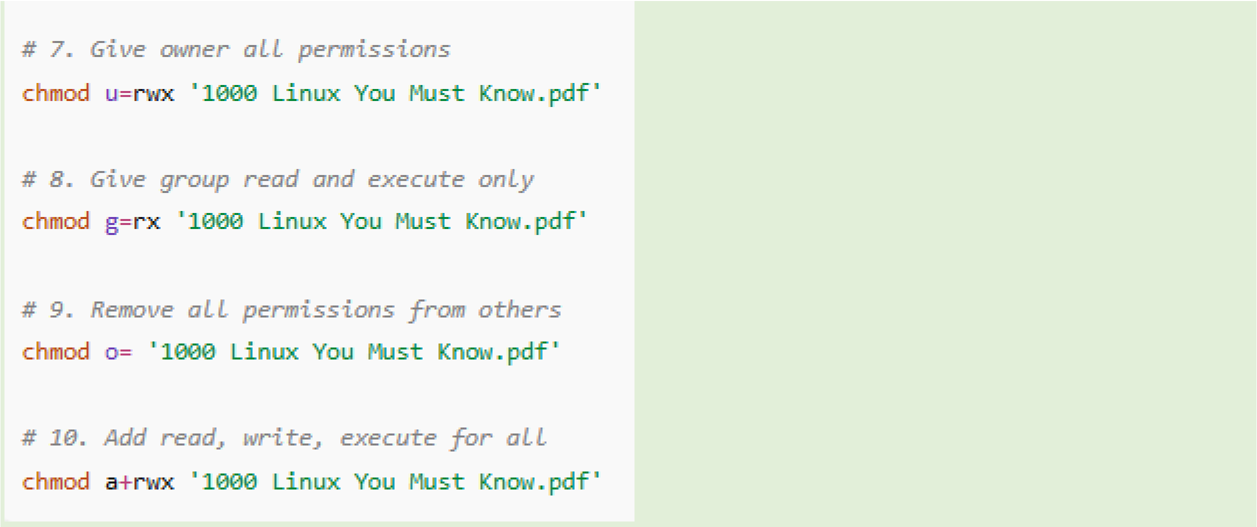
Instead of numbers, Linux lets you add/remove permissions using letters:

- r = read
- w = write
- x = execute
- u = owner/user
- g = group
- o = others
- a = all (owner+group+others)
- + = add permission
- - = remove permission
- = = set exact permission

Examples



```
# 1. Add execute for owner
chmod u+x '1000 Linux You Must Know.pdf'

# 2. Add write for group
chmod g+w '1000 Linux You Must Know.pdf'

# 3. Add read for others
chmod o+r '1000 Linux You Must Know.pdf'

# 4. Remove write for owner
chmod u-w '1000 Linux You Must Know.pdf'

# 5. Remove read for group
chmod g-r '1000 Linux You Must Know.pdf'

# 6. Remove execute for others
chmod o-x '1000 Linux You Must Know.pdf'
```



```
# 7. Give owner all permissions
chmod u=rwx '1000 Linux You Must Know.pdf'

# 8. Give group read and execute only
chmod g=rx '1000 Linux You Must Know.pdf'

# 9. Remove all permissions from others
chmod o= '1000 Linux You Must Know.pdf'

# 10. Add read, write, execute for all
chmod a+rwx '1000 Linux You Must Know.pdf'
```

REVISON QUIZ

Part 1 – Numeric/Binary

1. The current permission of the file is -rw-r--r--.
   Question: What numeric permission should you use with chmod to give owner full access, group read/write, others no access?

2. You want the file to have permissions:

   ○ Owner: read & write

- Group: read only
- Others: execute only
  Question: What is the numeric value for this permission?

3. The file currently has permission 664.
   Question: What command removes all owner permissions using numeric method?

Part 2 – Symbolic/Human-Readable

4. Question: Write a chmod command to add execute for owner only.
5. Question: Write a chmod command to remove write permission from group.
6. Question: Set the file so that owner has all permissions, group has read & execute, others have none using symbolic method.
7. Question: Remove all permissions from others using symbolic method.

Part 3 – True/False

8. chmod 777 filename gives all permissions to everyone.
9. chmod u-x filename removes execute permission from owner.
10. chmod a-w filename removes write permission from owner only.