🛡️ Linux Permission Management Summary

Linux permissions are essential for controlling who can access or modify files and directories. This system ensures secure and efficient collaboration among users and groups.

📁 Users, Groups & Ownership

- **Users**: Every file has an owner.
- **Groups**: Every file is associated with a group.
- **Permissions** are defined for: **Owner**, **Group**, and **Others**.

Example:

```
drw-rw-r-- 3 cry0l1t3 cry0l1t3 4096 Jan 12 12:30 scripts
```

🔑 Permission Types

Symbol	Meaning
r	Read
w	Write
x	Execute / Traverse (for directories)

To **traverse** a directory, execute (x) permission is required.

❌ Permission Denied Example

```
ls -al mydirectory/
ls: cannot access 'mydirectory/script.sh': Permission denied
```

Even if read permission exists, **execute permission** is required to access directory contents.

🧮 Octal & Binary Permission Representation

Binary Notation	Binary	Octal	Permission
4 2 1	111	7	rwx
4 2 1	101	5	r-x
4 2 1	100	4	r--

Octal example:

```
chmod 754 shell
-rwxr-xr--  (Owner: rwx=7, Group: r-x=5, Others: r--=4)
```

🔧 Changing Permissions

```
chmod a+r shell           # Add read permission for all users
chmod 754 shell           # Set exact permissions using octal
```

👥 Changing Ownership

```
chown <user>:<group> <file>
chown root:root shell
```

🚀 SUID & SGID

- **SUID (`s`)**: Executes with the owner's permissions.
- **SGID (`s`)**: Executes with the group's permissions.
- Appears as `s` instead of `x` in file permissions.

> ⚠️ **Security Risk**: Improper use can lead to privilege escalation.

Example:

```
-rwsr-xr-x 1 root root ... someprogram
```

🔒 Sticky Bit

- **Sticky Bit (`t`)**: Only the owner or root can delete/rename files in a shared directory.
- Appears as `t` or `T` at the end of directory permissions.

```
drw-rw-r-t 3 cry0l1t3 cry0l1t3 ... scripts   # Executable + sticky
drw-rw-r-T 3 cry0l1t3 cry0l1t3 ... reports   # Not executable + sticky
```

> Useful in `/tmp` and other shared directories to prevent unauthorized deletion.

✅ Summary

- Use `r`, `w`, `x` for Read, Write, Execute permissions.
- Modify with `chmod`, `chown`, `chmod 754`, `chmod a+r`, etc.
- Understand **SUID**, **SGID**, and **Sticky Bit** for advanced permission control.
- Practice by analyzing file permission strings and converting between symbolic, binary, and octal forms.

Tip: Use `ls -l`, `stat`, and `chmod --help` for insights into file permissions.