Linux User Management, Permissions, and Special Permissions

Introduction to User Types:

- In Linux, users are categorized into three types: Admin users with unlimited power (UID 0), General users with limited power (UID > 1000 – 60000), and Service or System users used by processes (UID 1-999 or > 60000 – 65335).

User Creation:

- Users can be created for various purposes such as testing, deploying applications, or running programs.
- To check the location of the "python" command, use the "which python" command.

Accessing User Information:

- To view user information, open the "/etc/passwd" file using the command "vim /etc/passwd."
- The user "umesh" is dedicated to running Python programs.

User Creation Commands:

- To create a user "pop1111" with a specific UID (1234), use the command "useradd –u 1234 pop1111."
- Create a user "pop222" with a specific shell "sh" using the command "useradd –s /bin/sh pop222."
- Create a user "pop3333" with a specific home directory "/tmp/pop3333" using the command "useradd –d /tmp/pop3333 pop3333."

User Modification:

- Use the "usermod" command to modify information for pre-created users.

Changing UID:

- To change the UID of user "pop3333" to "4444," use the command "usermod –u 4444 pop3333."

User Deletion:

- Delete the user "pop3333" using the command "userdel pop3333." To delete the home directory as well, use "userdel –r pop3333."

Userdel Command Options:

- To explore more options for the "userdel" command, use "userdel –h."

Useradd Command:

- The "useradd" command creates users and updates files like "/etc/passwd," "/etc/shadow," "/etc/group," etc.

/etc/shadow File Overview:

- The "/etc/shadow" file contains user information with nine fields separated by colons.
- Fields include username, password, last password change date, minimum and maximum password age, warning, password inactive, account expiry, and a reserved field.

Setting User Password:

- Use the "passwd" command to set the password for a user, e.g., "passwd rahul."
- Passwords are stored in hash format (SHA512) in the "/etc/shadow" file.

Removing, Locking, and Unlocking Passwords:

- Remove the password for user "rahul" with "passwd –d rahul."
- Lock the password of user "rahul" using "usermod –L rahul."
- Unlock the password of user "rahul" with "usermod –U rahul."

Chage Command:

- Use the "chage" command to get human-readable information from "/etc/shadow."

  chage -l username

Exploring /etc/login.defs:

- Default login information is stored in the "/etc/login.defs" file.

Password Restriction:

- Restrict user "rahul" from changing the password for the next 5 days by specifying in the fourth field of the "/etc/shadow" file.

Setting Warning Days:

- Change the number of warning days to 50 before password expiry for user "rahul" using "chage –W 50 rahul."

Additional Resources:

- To view the manual of the "/etc/shadow" file, use the command "man 5 shadow."

Introduction to User Interaction:

- Interacting with the operating system involves running programs, requiring user access to files and directories.
- User permissions are crucial for setting up access levels for non-root users.

Types of Permissions:

- For files, there are three types of permissions (modes): Read (r), Write (w), and Execute (x).
- Permissions on directories include Read (ls), Write (touch, rm), and Execute (cd).

Checking File Details:

- Use the command `ls -l /etc/passwd` to view file details.
- Example: `-rw-r--r--` indicates a file with read and write permissions for the owner, and read-only for others.

Directory Permissions:

- Directory permissions include Read (ls), Write (touch, rm), and Execute (cd).
- Use the command `ls -l -d /etc/` to view directory details.

Restricting Access to /root/:

- Prevent other users from accessing /root/ using appropriate permission settings.

Changing Permission Categories:

- Permissions are categorized into User owner (u), Group owner (g), and Other users (o).
- When a user creates a file or directory, they become the owner.

Changing Folder Ownership:

- Only the root user can change the ownership of /root/. Use `chown vimal /root/` to change ownership to user "vimal."

Modifying Other User Permissions:

- Add execute permission for other users with `chmod o+x /root/`.
- To add read or write permissions, use `chmod o+r /root/` or `chmod o+w /root/` respectively.

Handling File Execution Permission:

- Deny execution permission for other users on a specific file, like /usr/bin/date.

- Example: `chmod o-x /usr/bin/date`.

Switching Users with 'su' Command:

- Use the `su -` command to switch to a user and land in their home directory.

Changing Group Ownership:

- Change the group ownership of /code/ to user "vimal" using `chown vimal /code`.

Creating Users and Groups:

- Create users "yash," "sarah," and "raj" with `useradd` command.
- Create a group "lwgroup" using `groupadd lwgroup`.

Understanding /etc/group File:

- The /etc/group file has four fields: Group name, Group password (linked to **/etc/gshadow), GID (group identifier), and Members.**

Adding Users to a Group:

- Add users "yash," "sarah," and "raj" to the group "lwgroup" using `usermod -G lwgroup <username>`.

Group Ownership and Permissions:

- Change the group ownership of /code/ to "lwgroup" using `chgrp lwgroup /code`.
- Modify group permissions on /code/ using `chmod g+rwx /code`.

Setting Permissions with Numbers:

- Permissions can be set using numbers: Read (4), Write (2), and Execute (1).
- Example: `chmod 754 /code` grants read, write, and execute to the owner, read and execute to the group, and read to others.

Introduction to Special Permissions:

- Special Permissions in Linux are advanced settings that provide unique functionalities to enhance security and control in file and directory management.
- Three types of Special Permissions: Sticky Bit, SGID (Set Group ID), and SUID (Set User ID).

SUID Permission:

- SUID stands for Set User ID, applicable to executable files.
- Users executing a file with SUID gain temporary privileges of the file's owner.
- Example: Enabling SUID on passwd allows regular users to change their passwords without administrative privileges.

SGID Permission:

- SGID, Set Group ID, is another special permission for executable files and directories.
- Users executing a file with SGID temporarily assume the group ownership of the file.
- SGID on directories ensures new files inherit group ownership from the parent directory, facilitating collaboration.

Sticky Bit:

- The Sticky Bit is a special permission applicable only to directories.
- It restricts the deletion or renaming of files within the directory to the file owner, directory owner, and superuser.
- Commonly used in shared directories to ensure users can only modify their files.

Implementing Sticky Bit in Shared Folder:

- In a shared folder like "/share," users in the group "lw" have all permissions.
- Without the sticky bit, users can create and delete each other's files.

Setting Sticky Bit:

- Use the command `chmod o+t /share/` to set the sticky bit on the shared folder.
- After setting the sticky bit, users are restricted from deleting files created by others.

Setting Specific Permissions with Numbers:

- Use the command `chmod 1570 /share` to set permissions.
- Example: Read and execute (5) for users, full permissions (7) for the group, and no permissions (0) for others.

User and Group Ownership:

- User and group ownership automatically set when a user creates a file or directory.
- Primary group assigned during user creation, and changing it requires root access.

Setting Group Password:

- Use the command `gpasswd lwgroup` to set a group password, updated in "/etc/gshadow."

Changing Primary Group:

sudo usermod -g new_primary_group username

newgrp team:-This command would start a new shell session with the primary group set to "team". It allows the user to work in a new environment where the

primary group is temporarily changed to "team". Any files or directories created during this session will have the group ownership set to "team" by default.

SGID on Shared Folder:

- Set SGID and sticky bit on a shared folder ("/share") using the appropriate command.
- chmod 2xxx directory

Use Cases of SUID:

- Exercise caution with SUID as it can potentially compromise system security.
- Setting SUID on critical executable files like "/usr/bin/cat" allows users to perform actions with elevated privileges.
- chmod 4xxx file

SUID on Passwd Command:

- SUID on the "/usr/bin/passwd" executable allows users to change passwords without directly accessing "/etc/shadow."