Linux Operating System and Applications

User and Group Management

User Definition

- Users are defined in a system to determine **"who can access what"** within that system.
- In Linux, each user has a unique identifier called a **UID (User ID)**:
  - 0 – 99: system or administrative users
  - > 99: regular or non-system users
  - >= 500: standard (normal) users
- Each user belongs to **at least one group**, and each group also has a unique identifier called a **GID (Group ID)**.

Types of Users in Linux

Root User (Superuser)

- UID: `0`
- Full control over the entire system.
- Can install/remove software, manage users, change system configurations, and access any file.
- Represented as `root`.

System Users

- UID: typically from `1` to `99` (or up to 999 depending on the distro).
- Used by system services and background processes (e.g., `daemon`, `bin`, `nobody`).
- Do not log in interactively.
- Help separate system-level processes for security and stability.

Regular (Normal) Users

- UID: `>= 1000` (or `>= 500` in older systems).
- Created by administrators or during OS installation.
- Used by people to log in and perform daily tasks (e.g., `john`, `alice`).
- Have limited permissions to ensure system security.

Service or Application Users

- Similar to system users but often created during the installation of specific applications (e.g., `mysql`, `nginx`).
- Have just enough permissions to run the specific service.

Files that store user-related information

- `/etc/passwd`: Contains user account information such as login name, encrypted password placeholder, UID, GID, home directory, and login shell.

  → Each line represents one user.

- `/etc/shadow`: Contains encrypted user passwords and password aging information (e.g., password expiration, minimum/maximum password age, etc.).

- `/etc/group`: Contains group information, including group name, GID, and list of users belonging to the group.

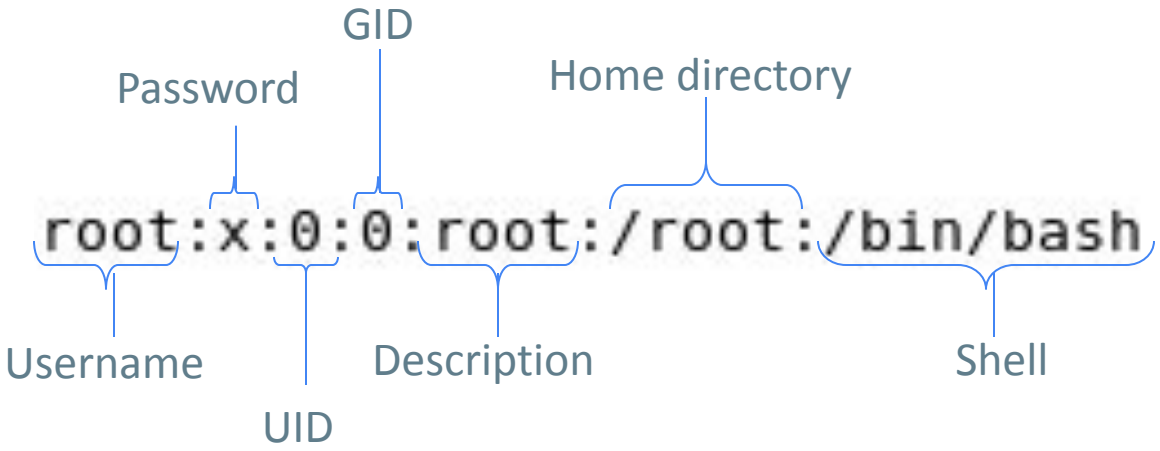
- `/etc/gshadow`: Contains group passwords in hashed form (rarely used).

The /etc/passwd file

- Contains essential information about user accounts.
- Each line represents a single user and includes details such as:
  - Username
  - Encrypted password placeholder (usually `x` if shadow passwords are used)
  - UID (User ID)
  - GID (Group ID)
  - User description or full name
  - Home directory
  - Default login shell
- This file is **world-readable**, but does **not** store actual password hashes (those are in `/etc/shadow`).

The /etc/passwd file

- Examples

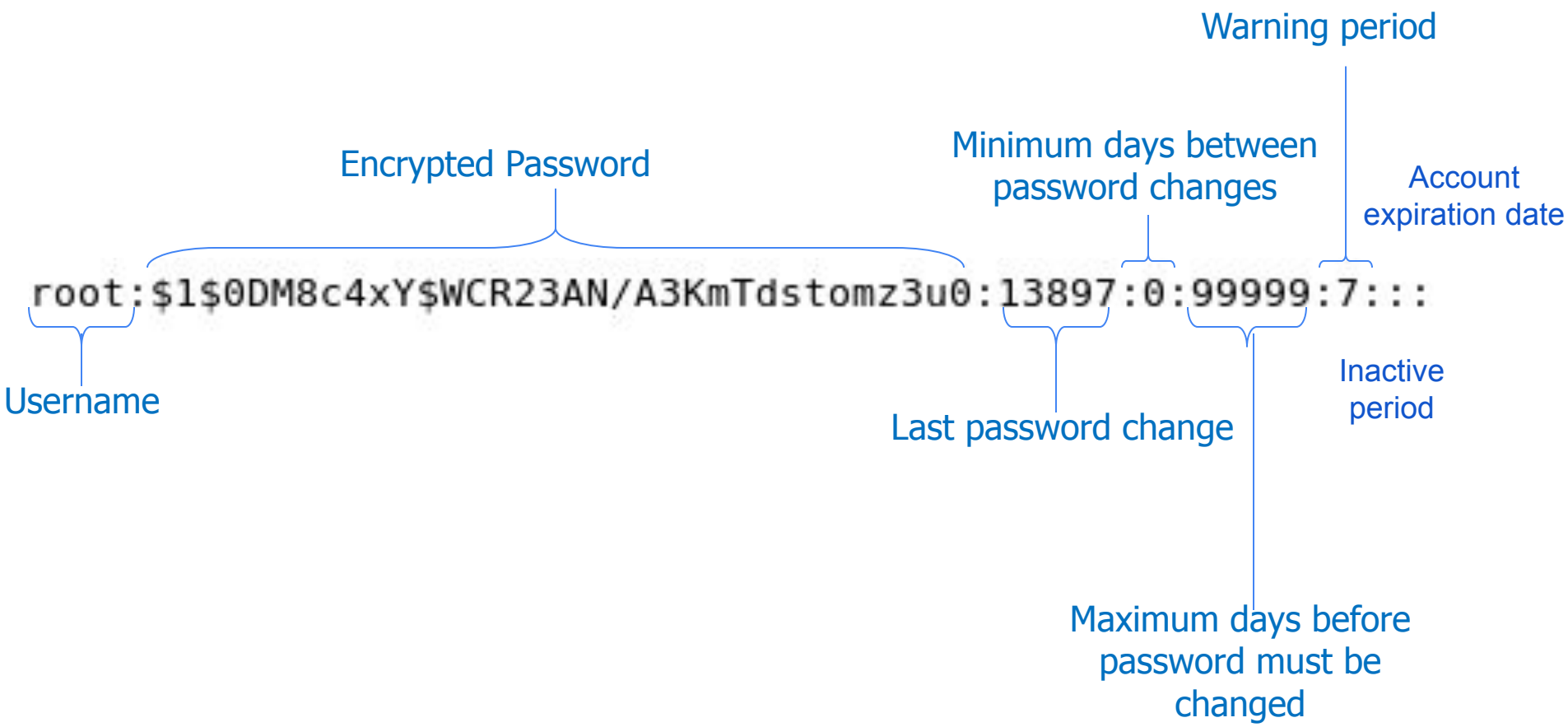


The /etc/shadow file

- Stores **secure password information** for user accounts.
- Each line corresponds to one user and contains fields related to:
  - **Encrypted password**
  - **Password aging and expiration policies**
  - **Account expiration**
- This file is **readable only by the root user** or processes with appropriate privileges, ensuring password hashes remain protected.

The /etc/shadow file

- Example



Password Policies

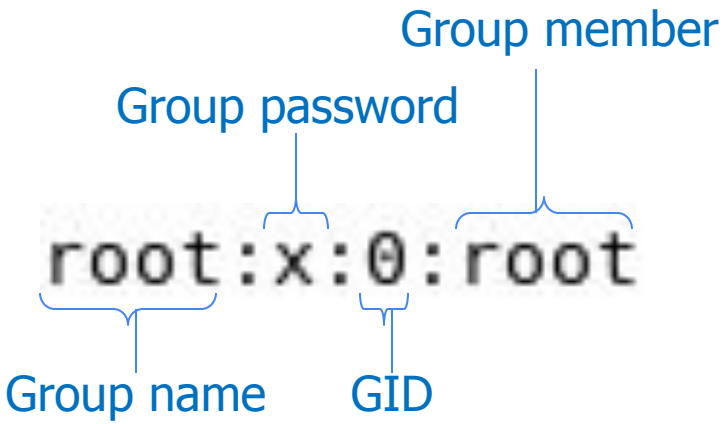
- **Username** – Must match a user in `/etc/passwd`
- **Encrypted password** – Hash of the password (or special values like `*`, `!` to disable login)
- **Last password change** – Days since Jan 1, 1970
- **Minimum days between password changes** – Prevents too-frequent changes
- **Maximum days before password must be changed**
- **Warning period** – Days before expiry to warn the user
- **Inactive period** – Days after password expiry before account is disabled
- **Account expiration date** – Absolute date (days since Jan 1, 1970)
- **Reserved field** – Currently unused

The /etc/group file

- Stores **group account information**.
- Each line defines a single group and includes:
  - Group name
  - Group password (rarely used)
  - Group ID (GID)
- List of members (users who belong to the group in addition to their primary group)
- This file is **world-readable** and works alongside `/etc/passwd` and `/etc/gshadow`.

The /etc/group file

- Example



User Management Tools

1. Command-Line Management:

- `useradd` – Create a new user
- `usermod` – Modify user information
- `userdel` – Delete a user (`-r` option deletes the user's home directory)
- `groupadd` – Create a new group
- `groupdel` – Delete a group
- `groupmod` – Modify group information
- `groups` – View group memberships

2. Graphical Interface Management:

- Use system settings or GUI tools (e.g., `Users and Groups` in desktop environments)

3. Direct File Editing:

- Manually edit system files:
  - `/etc/passwd`, `/etc/shadow`
  - `/etc/group`, `/etc/gshadow`

Default Configurations

- When using the `useradd` command **without options**, the new user will be created using **default settings**.
- **Files that define default user settings:**
  - `/etc/default/useradd` – General default settings for new users
  - `/etc/skel/` – Template directory: contents are copied to the new user's home directory
  - `/etc/login.defs` – Defines system-wide defaults like UID ranges, password aging, etc.
- To **change default behavior**, modify these files directly.

Switch Users

- The `su` (substitute user) command is used to **switch to another user account**.
- **Syntax:** su [- or -l] username
  - **-, -l:** Starts a **login shell**, executing the target user's environment and login scripts.
  - If `username` is omitted, it defaults to **`root`**.
- To return to the previous user, use the `exit` command.

Switch Users Examples

```
# Switch to another user (e.g., bob)
~$ su bob
Password:           # Enter bob's password

# Switch to root (default if no username is given)
~$ su
Password:           # Enter root's password

# Switch to root with full login environment
~$ su -             # or su -l

# Prompt will change to # (indicates root)
~# whoami
root

# Exit to return to the previous user
~# exit
exit
~$
```

Changing Passwords

- To change a user's password, use the `passwd` command:

```
# passwd henry
current password :
new password:
retype new password:
```

- Password selection tips:
  - Avoid using **dictionary words** or **names**
  - Use a mix of **letters** and **digits**
  - Include **symbols** such as**: !**, **@**, **#**, **$**, **%**, etc.
- **Do not allow guest accounts** to log in to the system

Changing Password Expiration Settings

- Use the `chage` command to manage password expiration policies:

  chage [options] <user>

- Common options:
  - `-m <mindays>` — Minimum number of days between password changes
  - `-M <maxdays>` — Maximum number of days the password is valid
  - `-d <lastdays>` — Set the date of the last password change
  - `-I <inactive>` — Number of days after password expiration before the account is locked
  - `-E <expiredate>` — Account expiration date (format: `YYYY-MM-DD` or `MM/DD/YY`)
  - `-W <warndays>` — Number of days before expiration to warn the user

Account Security

- Set an expiration date for temporary accounts:

```
# usermod –E 2003-12-20 henry
```

- Lock inactive accounts (e.g., lock after 5 days of password expiration):

```
# usermod –f 5 henry
```

- Find and delete all files/directories owned by a user outside their home directory:

```
# find / -user henry –type f –exec rm –f {} \;
# find / -user henry –type d –exec rmdir {} \;
```

⚠️ Be cautious when running these commands, especially on production systems.

Privilege Delegation Policy

Linux is a multi-user, multi-administrator environment.

- If everyone uses the **root account**, it becomes **difficult to trace** who made specific system changes.
- Each user should operate under their **own account**.
- When elevated privileges are required, users should **temporarily "borrow" root privileges** using the sudo command.

This approach improves **security**, **accountability**, and **auditability**.

Using sudo

- When using sudo, the user (e.g., sv) is prompted to enter **their own password** (not the **root** password).
- Upon successful authentication, the user can execute commands with elevated privileges.

sv@localhost:~

```
[sv@localhost ~]$ service vsftpd restart
Shutting down vsftpd:                                      [FAILED]
Starting vsftpd for vsftpd: 500 OOPS: cannot open config file:/etc/vsftpd/vsftpd
.conf
                                                           [FAILED]
[sv@localhost ~]$ sudo service vsftpd restart

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for sv:
Sorry, try again.
[sudo] password for sv:
Shutting down vsftpd:                                      [  OK  ]
Starting vsftpd for vsftpd:                                [  OK  ]
```

Using sudo examples

```
# Update system packages
sudo yum update

# Install a new package
sudo yum install httpd

# Start the Apache (httpd) service
sudo systemctl start httpd

# Enable httpd to start at boot
sudo systemctl enable httpd

# Edit the firewall configuration
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --reload

# Switch to user 'postgres'
sudo -u postgres psql

# Reboot the system
sudo reboot
```

Who Can Use sudo?

- The users allowed to run `sudo` commands and their permitted privileges are defined in the `/etc/sudoers` file.
- To edit this file safely, use the `visudo` command with **root** privileges.
- `visudo` works like the `vi` editor but is specially designed for editing the `sudoers` file, preventing syntax errors and handling variations across different Linux distributions.
- sudoers File Syntax

  username/group servername = (usernames_to_run_as) command

  - The **`usernames_to_run_as`** field is optional; if omitted, the command runs as **root** by default.
  - Multiple usernames or commands can be listed, separated by commas (`,`).

Conventions in sudoers File

- For groups, prepend the group name with a `%` sign in the **first column**.
- Use the keyword **`ALL`** to represent all users, all hosts, or all commands (examples can be provided).
- If a line is too long, use a backslash `\` at the end of the line to continue onto the next line.
- If the sudoers file is only used on a local machine, the **hostname field** is usually set to `ALL`.

sudoers File Examples

```
# Allow user 'alice' to run all commands as root on any host
alice ALL=(ALL) ALL

# Allow group 'admins' to run all commands as any user on any host
%admins ALL=(ALL) ALL

# Allow user 'bob' to run specific commands only
bob ALL=(ALL) /usr/bin/systemctl, /usr/bin/journalctl

# Allow user 'carol' to run commands as user 'backup' on localhost only
carol localhost=(backup) /usr/bin/rsync

# Long line example, continued with backslash
dave ALL=(ALL) /usr/bin/command1, /usr/bin/command2, \
        /usr/bin/command3

# Allow all users on host 'server1' to run all commands as root
ALL server1=(ALL) ALL
```

Q&A