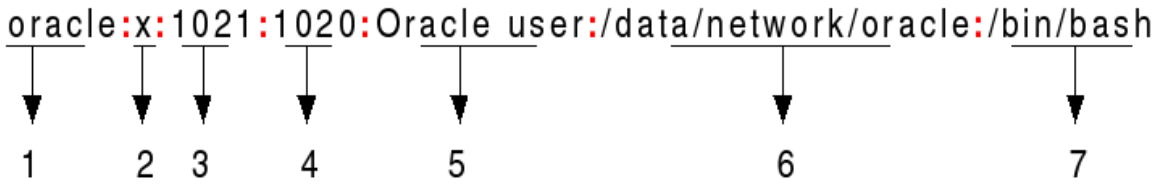
Name	**Chinmay R Mhatre**
Roll No.	**22102B2001**
Division	**CMPN B**
Batch	**4**
Subject	**Cryptography and System Security**
Task No.	**3**

Title: /etc/passwd vs /etc/shadow file system

❖ etc/passwd:

The "/etc/passwd" file stores essential information required during login. In other words, it stores user account information. The /etc/passwd is a plain text file. It contains a list of the system's accounts, giving for each account some useful information like user ID, group ID, home directory, shell, and more. The /etc/passwd file should have general read permission as many command utilities use it to map user IDs to usernames. However, write access to the /etc/passwd must only limit for the superuser/root account.

`/etc/passwd` is a plain text file on Unix-like operating systems (such as Linux) that stores essential information about user accounts. Each line in the file represents a single user account and contains several fields separated by colons (:). These fields typically include:Generally, /etc/passwd file entry looks as follows:

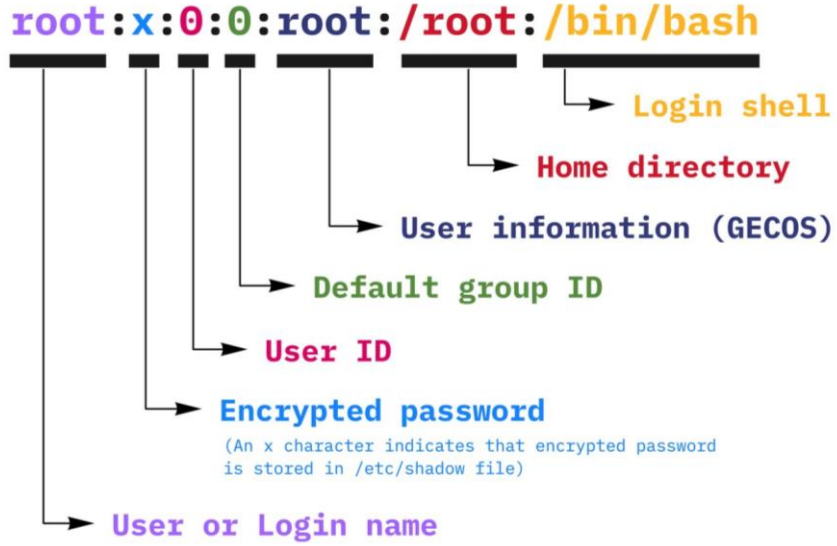


1. /etc/passwd file format

1. **Username**: This field represents the name used by a user to log in to the system. It should be between 1 and 32 characters in length. Each user on the system must have a unique username.
2. **Password**: In the past, this field used to store the encrypted password of the user. However, for security reasons, modern Unix-like systems store an 'x' character in this field, indicating that the encrypted password is stored in the /etc/shadow file. The /etc/shadow file is readable only by the root user, enhancing the security of password storage.
3. **User ID (UID):** Every user on the system is assigned a unique numerical identifier known as the User ID (UID). UID 0 (zero) is reserved for the root user, which has administrative privileges. UIDs 1-99 are often reserved for predefined system accounts, and UIDs 100-999 are typically used for administrative and system accounts or groups.
4. **Group ID (GID):** This field specifies the primary group ID of the user. The primary group is defined in the /etc/group file. Each user is associated with a primary group, and this field contains the numerical identifier (GID) of that group.
5. **User ID Info (GECOS):** The GECOS field is also known as the comment field. It allows additional information about the user to be stored, such as the user's full name, contact

information, or any other relevant details. This information can be accessed using commands like finger.

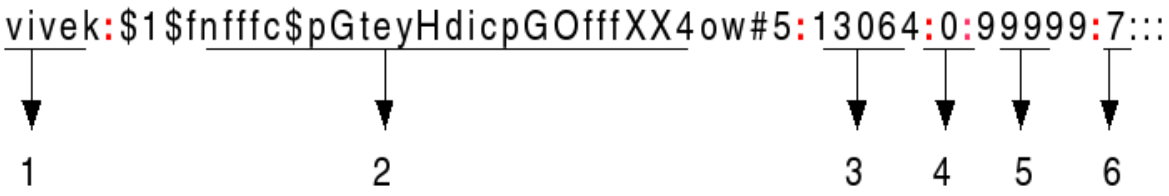
6. **Home Directory:** This field specifies the absolute path to the user's home directory. Upon logging in, the user is placed in this directory by default. If the specified home directory does not exist, the user's directory becomes the root directory ('/'). The home directory is where users store their personal files and configurations.
7. **Command/Shell:** This field specifies the absolute path to the user's default shell or command interpreter. When a user logs in, the system executes this shell, providing the user with a command-line interface to interact with the system. Common shells include /bin/bash, /bin/sh, /bin/zsh, etc. Additionally, instead of a shell, this field can contain a command like /sbin/nologin, which prevents direct login by terminating the session immediately after authentication. This is often used for service accounts or users who should not have interactive shell access.



Understanding the structure and content of the /etc/passwd file is crucial for managing user accounts and permissions on Unix-like systems. It provides essential information about each user, facilitating user authentication and access control.

❖ etc/shadow:

The /etc/shadow file is a critical component of Unix-like operating systems, such as Linux. It is primarily responsible for storing secure user authentication data, specifically the hashed passwords for user accounts. Unlike the /etc/passwd file, which traditionally stored password hashes directly, the /etc/shadow file enhances security by ensuring that only privileged users, typically the root user, can access it. All fields are separated by a colon (:) symbol. It contains one entry per line for each user listed in /etc/passwd file. Generally, shadow file entry looks as follows (click to enlarge image):



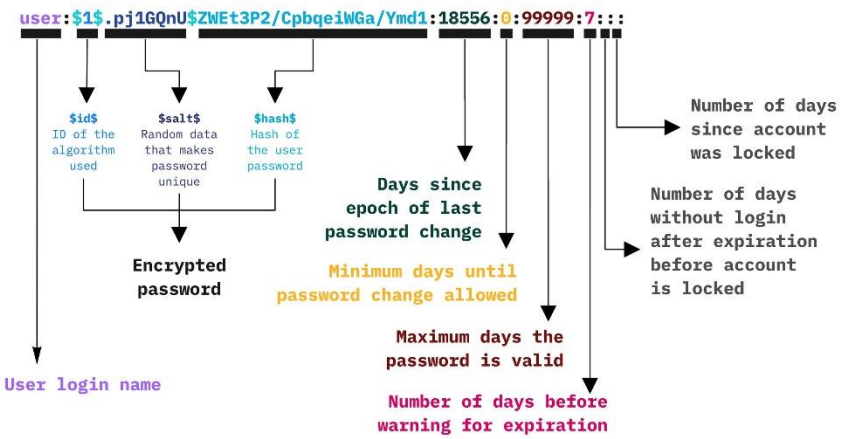
2. /etc/shadow file format

As with the /etc/passwd, each field in the shadow file is also separated with ":" colon characters as follows:

1. **Username** : This field corresponds to the username of the user account. It is the same username found in the /etc/passwd file and is used for user authentication.
2. **Password** : Instead of storing plaintext passwords, the /etc/shadow file stores hashed passwords. The actual password hash is stored in this field. The password should be minimum 15-20 characters long including special characters, digits, lower case alphabetic and more. Usually password format is set to `$id$salt$hashed`, The `$id` is the algorithm prefix used On GNU/Linux as follows
    1. **`$1$`** is MD5
    2. **`$2a$`** is Blowfish
    3. **`$2y$`** is Blowfish
    4. **`$5$`** is SHA-256
    5. **`$6$`** is SHA-512
    6. **`$y$`** is yescrypt
3. **Last password change (lastchanged)** : The date of the last password change, expressed as the number of days since Jan 1, 1970 (Unix time). The value 0 has a special meaning, which is that the user should change her password the next time she will log in the system. An empty field means that password aging features are disabled.

Minimum Password Age (min): This field specifies the minimum number of days that must pass before the user can change their password again. It helps enforce password policies by preventing users from changing their passwords too frequently.

4. **Maximum Password Age (max):** This field indicates the maximum number of days a password is valid before it must be changed. It enforces password expiration policies, ensuring that passwords are regularly updated for security reasons.
5. **Password Warning Period (warn):** This field specifies the number of days before password expiration that the user will receive a warning message.
6. **Password Inactivity Period (inactive):** If a password expires, the account may be disabled after this number of days of inactivity.
7. **Account Expiration Date (expire):** The date of expiration of the account, expressed as the number of days since Jan 1, 1970.



❖ Difference between etc/passwd and etc/shadow

Feature	/etc/passwd	/etc/shadow
Purpose	Stores basic user account information.	Stores encrypted passwords and security data.
Accessibility	Readable by all users.	Accessible only by privileged users.
Contents	Usernames, UIDs, GIDs, home directories, shells.	Encrypted passwords, password aging parameters.
Security	Less secure (historically stored passwords).	More secure (passwords stored in hashed form).
Password Handling	Historically stored encrypted passwords.	Stores hashed passwords and security parameters.
Usage	Basic user information and system utilities.	Enhanced password security and management.

❖ Why is shadow used even when passwd is available?

1. The /etc/shadow file is used alongside the /etc/passwd file to make sure our passwords are super safe. See, back in the day, passwords used to be stored right in the /etc/passwd file, but that wasn't very secure because if someone got access to that file, they could see everyone's passwords!
2. So, smart people came up with the idea of the /etc/shadow file. It's like a super-secret vault just for passwords. Instead of storing passwords directly, it stores something called "hashes," which are like secret codes that only the computer can understand. Even if someone gets into the /etc/shadow file, they can't figure out the passwords because they're all scrambled up!
3. Plus, only the really important computer users, like the root user, can even look inside the /etc/shadow file. That means our passwords are safe from regular users snooping around.
4. So, even though we have the /etc/passwd file with basic user info, we use the /etc/shadow file to keep our passwords extra safe and make sure nobody can mess with them.