User Management

/etc/passwd file

- It is the primary repository of users on a standalone system. It may be replaced by a centralized LDAP server in large environments.
- The file contains seven fields separated by colons:
  - Login name
  - Password placeholder
  - UID
  - GID
  - Human friendly information (real name, phone…etc.)
  - Home directory
  - Login shell

Login name

- It must be unique on the system
- It cannot contain colons (:) or the newline character; as these have meanings in the passwd file
- If an NIS is used, the login name is limited to eight characters long
- Linux allows up to 32 characters to form a login name. it also requires the first character to be a lower case letter or an underscore.
- Linux accepts characters from a to z, both lower and upper case, in addition to dashes (-) and underscores (_).

Login name best practices

- Should be no more than 8 characters for backward compatibility with legacy systems.
- Should be all in lower case for consistency
- Should not be nicknames; so as not to affect the organization’s image. Often nicknames are used for e-mails.
- It’s wise to use a naming scheme. For example first initial *dot* last name.

Password placeholder

- The password field in the `/etc/passwd`. The real encrypted password is stored in `/etc/shadow` file
- You can opt to use `/etc/passwd` for storing passwords (although not recommended) using `pwunconv` command. If you wan to revert to `/etc/shadow` use `pwconv`
- Linux supports a number of encryption algorithms. They can be set in the `/etc/login.defs` file, together with the password length.
- If you are manually editing the `/etc/passwd` and `/etc/shadow` files to manage users, you must put a star in the password placeholder to prevent the user from logging in unless a password is provided. Leaving the password field empty allows the user to login without entering a password.
- The encrypted string that starts with $ means that it was not encrypted using DES. For example, 1 means an MD5-based algorithm was used, while 6 indicates a SHA512 algorithm.
- If an exclamation mark is placed before the password string this means that the account is locked
- If two users chose the same password, that does not mean that you will find the same encrypted string because Linux adds a “salt” string to the password before encryption.

UID – User Identification

- Users are defined by their names only for the benefit of users. All applications and filesystems use the UID to identify users on Linux.
- The root has UID of 0.
- Linux may create *fake* accounts to be owners of daemons and services like daemon, bin, mail…etc.
- Although the system permits having multiple users with the same UID, it is highly not recommended, especially for the root account.
- It is advised to have the unique UID's across the entire network. This ensures that every person has the same UID on any system. Such a control will mitigate potential security vulnerabilities when sharing files, like through NFS for example.
- As the number of connected systems gets higher, it will be hard to keep this unique UID control; so it's advisable to use a centralized login system like LDAP.

GID – group identification

- Used to identify groups by number
- Only the primary group is listed
- The root group GID is 0
- System users are also assigned to groups
- They are mainly used to share files among users
- If setgid is used, any file created in the directory will bear the group id of this directory instead of it’s own

GECOS

- It stands for “the General Electric Comprehensive Operating Supervisor”, which is a brand of mainframe computers.
- This field contains human readable, personal information about the user. For example, full name, office number…etc.
- You can view information about a user by using the finger command
- You can use the chfn to update this information (will not work on LDAP)
- If you want to add it manually, make sure you separate the fields by commas (,). Of course you’d do this as root

Home directory

- The default directory to which the user is put upon logging in
- It defaults to /home followed by a directory that has the same name as the user's. /home can be changed in /etc/default/useradd
- It contains environment files like `.bash_profile` and `.bash_rc`
- Sometimes administrators create users' home directories as network shares on a central server to provide *roaming profiles* for the users. That is, the user will have the same environment regardless of the machine used for logging
- If there is no home directory specified for the user, an error message will be displayed and the user will be placed on the root directory /

The login shell

- It is the command line interpreter
- It defaults to BASH in Linux. Other shell interpreters are available including ksh and tch.
- BASH stands for Bourne Again Shell, which is the Bourne Shell (sh) successor.
- The sh and csh are just links to the modern tcsh and BASH shells
- Users may be allowed to change their login shells with `chsh` command
- The changed shell should be one of the shells in /etc/shells. The administrator can edit this file
- The administrator can change the login shell of the user by directly editing /etc/passwd file using `vipw` command

The /etc/shadow file

- Used to store the encrypted passwords
- Both /etc/shadow and /etc/passwd must co-exist for user management. Tools like `useradd` and `usermod` are used to manage both of them.
- The file contains the following nine fields:
  - Username
  - Encrypted password
  - The date of last password change
  - Minimum number of days allowed before a user can change the password
  - Maximum number of days allowed before a user can change the password
  - Number of warning days before the password expires
  - Number of days before the account gets disabled *after* a password has expired
  - Account expiration date. It represents the number of days sicne 1/1/1970 (this is not UNIX timestamp that is calculated as the number of seconds since 1/1/1970). If left blank, the account will never expire.

The /etc/group file

- It contains the current groups on the system and the users listed in each one
- Each group is represented by a line. Fields are separated by colons, no spaces are allowed
- The line contains the following fields:
  - Group name
  - Encrypted password or a placeholder
  - GID
  - Member users
- The password placeholder is used if the group has a password set (using `gpasswd` command). However, it is highly unusual.

Creating new users

- The `useradd` command is used to add new users to the system. It does the following tasks:
  - Adds a user entry to the /etc/passwd file
  - Adds a new group with the same name as the login name to the /etc/group file
  - Creates a new directory with the same name as the login name to /home directory
  - Sets the appropriate ownership and permissions to the home directory
  - Sets the user's mail directory and creates a mail alias
- Users can be added using, instead, the `vipw` and `vipw –s`, which would edit the /etc/passwd and /etc/shadow files respectively
- You must assign a password for the new user before he/she can use the account. Passwords are assigned using the `passwd` *`loginame`* command

Environment files

- Every shell has one or more environment files. Those are responsible for setting important session-wide variables.
- The following are examples of startup files:
  - .bash_profile and .bashrc for the BASH shell
  - .profile for the Korn shell
  - .vimrc for customizing the vim environment
- Those startup files are placed in /etc/skel directory. They get copied to the user's home directory upon account creation
- The /etc/profile file get's executed before any startup files are. It is a good place to store system-wide environment variables. However, this file can be easily overridden by the users

The `useradd` command

- It uses the the files `/etc/login.defs` and `/etc/default/useradd`
- In Ubuntu, the `adduser` command is a Perl script that provides more features than `useradd`. It uses the `/etc/adduser.conf` as a configuration file.
- The switch –D prints the defaults that useradd uses when creating new users.
- It adds an x in the /etc/passwd password field, and a !! In the /etc/shadow file, indicating that the user account is locked until the superuser assigns a default password.
- Appropriate entries are added in the /etc/group file

Useradd options

- You can use command line switches with the `useradd` command to customize user creation process. The most common are the following:
  - -c specify the friendly name for the user (often the full name)
  - -d a different home directory than the default
  - -g a different primary group than the default
  - -G add the user to a supplementary group
  - -s specify a login shell
  - -m move files from home directory to a new one
- Those same options can be used with the `usermod` command

Adding multiple users in one go

- Bulk user accounts can be created using the `newusers` command
- The command takes a file that contains the user accounts as it's argument
- The file is formatted the same way as the /etc/passwd file, except that the x placeholder in the password field is replaced with the actual password, in clear text.
- The newusers command does not copy the startup files from /etc/skel directory. Those files have to be manually created
- The user's file can be created using a script that will guarantee creating strong passwords. This file should be highly secured, and removed once the creation process is complete.

Deleting user accounts

- Users can be deleted using the `userdel` command.
- In Ubuntu there is the `deluser` script, which is a wrapper to `userdel`.
- [Ubuntu] It uses the configuration file /etc/deluser.conf to provide the following features (they can be turned on or off):
    - Remove the user’s home directory
    - Backup the user’s files
    - Remove all the files owned by the user
    - Delete the user’s group if empty
- You must make sure that a deleted user has been removed from the crontab jobs, and that any processes that is still running under his account is taken care of

Users’ locking and unlocking

- You can prevent a user from logging in (locking) using `usermod -L` *`login`*. You can unlock the account using `usermod -U` *`login`*
- Locking the account means putting an asterisk (*) or an exclamation mark before the encrypted password field in the /etc/shadow file as this destroys the hash. Unlocking the password removes those characters.