Linux System Administration-I
CSE-4043

Chapter 7: Adding New Users

NIBEDITA JAGADEV

Department of CSE

Asst. Professor

SOA Deemed to be University, Bhubaneswar, Odisha , India

nibeditajagadev@soa.ac.in

Contents

- Adding New Users
- /etc/passwd File
- Shadow password files
- Setting Password Restrictions

Adding New Users

- User as an entity - username(UID), GID.
- UID - typically a number for system to identify the user.
- GID – a number that recognizes a set of users that have similar functions.
- UID and GID together determine any user's access rights to files and system resources.
- Password file - /etc/passwd file that stores user account information.
- Password file is readable by all but writable only by root.

Adding New Users

- Assign username, uid, primary group.
- Enter this information in /etc/passwd.
- Assign a password.
- Set user account parameters, password aging, account expiration date etc.
- Create a home directory – users organization on the hard disk.
- Place Initialization files in the user's directory.
- Use chown and chgrp to make the user the owner of his home directory and initialization files.

Adding New Users

- Add the user to other facilities like disk quota system, mail system, printing system etc. Site-specific initialization tasks. Test the new account

/etc/passwd File

- The **/etc/passwd** file is a list of users recognized by the system
- Each line in the file represents one user and contains seven fields separated by colons:
- Login name
- Encrypted password placeholder
- UID (user ID) number
- Default GID (group ID) number
- “GECOS” information: full name, office, extension, home phone
- Home directory
- Login shell

- Login name
- Encrypted password placeholder
- UID (user ID) number
- Default GID (group ID) number
- “GECOS” information: (full name, office, extension, home phone)
- Home directory
- Login shell

/etc/passwd File

- UID : Unique identification numbers. Conventionally UIDs less than 100 are used for system accounts.
- GID : Determines user's primary group membership. This corresponds to a group in the /etc/group file. Conventionally GIDs less than 10 are used for system groups.
- User info : Only field in the entry that can have space. Also called as GECOS field. Distinct items within this field are separated by commas.
- Home-dir : Initial working directory.

/etc/passwd file (contd)

- Shell: The program that UNIX will use as a command interpreter for this user. This program is executed whenever the user logs in.
- Instead of manually editing the /etc/passwd file, it is preferable to have commands to add new user, as more than one scattered files might need to be changed. With commands all the necessary file updates can be ensured

Shadow password files

- A shadow password file is readable only by the super user and serves to keep encrypted passwords safe from prying eyes and password cracking programs.

Shadow password files

/etc/shadow contains one line for each user. Each line contains nine fields, separated by colons:

- Login name
- Encrypted password
- Date of last password change
- Minimum number of days between password changes
- Maximum number of days between password changes
- Number of days in advance to warn users about password expiration
- Linux: Days after password expiration that account is disabled
- Solaris/HP-UX: Days before account automatically expires
- Account expiration date
- A reserved field that is currently always empty, except on Solaris

Shadow password files

- The login name is the same as in **/etc/passwd**. This field connects a user's **passwd** and **shadow** entries.
- The encrypted password is identical in concept and execution to the one previously stored in **/etc/passwd**; a fake Solaris MD5 password is shown.
- The last change field records the time at which the user's password was last changed. This field is filled in by the **passwd** command.
- The fourth field sets the number of days that must elapse between password changes. The idea is to force authentic changes by preventing users from immediately reverting to a familiar password after a required change. However, we think this feature could be somewhat dangerous when a security intrusion has occurred. We suggest setting this field to 0.

Shadow password files

• The fifth field sets the maximum number of days allowed between password changes. This feature allows the administrator to enforce password aging; see page 906 for more information. Under Linux, the actual enforced maximum number of days is the sum of this field and the seventh (grace period) field.

• The sixth field sets the number of days before password expiration that **login** should begin to warn the user of the impending expiration.

• Solaris and HP-UX differ from Linux in their interpretation of the seventh

The sixth field sets the number of days before password expiration that **login** should begin to warn the user of the impending expiration.

Shadow password files

- The eighth field specifies the day (in days since Jan 1, 1970) on which the user's account will expire. The user may not log in after this date until the field has been reset by administrator. If the field is left blank, the account will never expire.
- The ninth field is reserved for future use. Linux and HP-UX honor this use, but Solaris uses the last 4 bits to count failed login attempts.

Setting Password Restrictions

- Specifying MINIMUM and MAXIMUM password lifetimes.
- Specifying the minimum password length.
- An entry in the shadow password file :

Name:coded-passwd:last_changed:min-days:max_days:warn_days:inactive_days:expire_date:

- Passwd command can also be used to specify password aging parameters.

Linux System Administration-I
CSE-4043

Chapter 7: Adding New Users

NIBEDITA JAGADEV

Department of CSE

Asst. Professor

SOA Deemed to be University, Bhubaneswar, Odisha , India

nibeditajagadev@soa.ac.in

Contents

- CREATING HOME DIRECTORY
- THE /ETC/GROUP FILE
- ADDING USERS: THE BASIC STEPS
- REMOVING USERS
- DISABLING LOGINS
- CENTRALIZING ACCOUNT MANAGEMENT
- LDAP and Active Directory

Creating home directory

- Use mkdir command to create the directory in appropriate location, such as :

 # mkdir /home/gkalra1

- Login Initialization files: Files for the shell the account will run. For example, .profile file for Bourne shell.
- These files are used to perform tasks that only need to be executed upon login, such as : Setting search path, Setting default file protection(umask), setting terminal type and other environment variables.
- Shell Initialization files: Includes tasks that need to be executed whenever UNIX creates a new shell.

Creating home directory (contd)

- Shell initialization files can include tasks like setting shell variables, defining shell aliases. For example, C shell uses .cshrc as initialization file.
- SHELL INITIALIZATION FILES ARE EXECUTED BEFORE LOGIN INITIALIZATION FILES.
- X Initialization files.
- Setting File ownership : After copying the appropriate initialization files to the user's home directory, the owner of the files and directory is changed to be the user. This is done by **chown** command.

THE /ETC/GROUP FILE

- The **/etc/group** file contains the names of UNIX groups and a list of each group's members.

Each line represents one group and contains four fields:

- Group name
- Encrypted password or a placeholder
- GID number
- List of members, separated by commas (be careful not to add spaces)

REMOVING USERS

- Remove the user from any local user databases or phone lists.
- Remove the user from the **aliases** file or add a forwarding address.
- Remove the user's crontab file and any pending **at** jobs or print jobs.
- Kill any of the user's processes that are still running.
- Remove the user from the **passwd**, **shadow**,11 **group**, and **gshadow** files.
- Remove the user's home directory.
- Remove the user's mail spool.
- Clean up entries on shared calendars, room reservation systems, etc.
- Delete or transfer ownership of any mailing lists run by the deleted user.

ADDING USERS: THE BASIC STEPS

- Have the new user sign your policy agreement.
- Edit the **passwd** and **shadow** files to define the user's account.
- Add the user to the **/etc/group** file (not really necessary, but nice).
- Set an initial password.
- Create, **chown**, and **chmod** the user's home directory.
- Configure roles and permissions (if you use RBAC; see page 190).

For the user:

- Copy default startup files to the user's home directory.
- Set the user's mail home and establish mail aliases.

For you:

- Verify that the account is set up correctly.
- Add the user's contact information and account status to your database.

DISABLING LOGINS

Modifying a user's password simply makes logins fail. It does not notify the user of the account suspension or explain why the account no longer works. An alternative way to disable logins is to replace the user's shell with a program that prints an explanatory message and supplies instructions for rectifying the situation. The program then exits, terminating the login session.

CENTRALIZING ACCOUNT MANAGEMENT

- Users need the convenience and security of a single login name, UID, and password across the site. Administrators need a centralized system that allows changes to be instantly propagated everywhere.

LDAP and Active Directory

- LDAP is a generalized, database-like repository that can store user management data as well as other types of data. It uses a hierarchical client/server model that supports multiple servers as well as multiple simultaneous clients.

Thank You