LPIC-1 TRAINING COURSE

Topic 107: Administrative tasks

Contents

Objectives

- Able to add, remove, suspend and change user account
- Use ***cron*** or ***anacron*** to run jobs at regular intervals and to use ***at*** to run jobs at a specific time
- Localize a system in a different language than English

1. Manage user and group accounts

Overview

Learn to:

- Add, modify and remove users and groups
- Manage user/group info in password/group databases
- Create and manage special purpose and limited accounts

Understanding Users and Groups

- Username restriction:
  - Can be up to 32 character (try to limit to 8 characters)
  - Case-sensitive
  - Can contain letters, numbers, underscores (_) and periods (.). Must begin with a letter
    - Eg: ***45u*** is invalid, ***u45*** is fine
- Group: means of organizing collections of user
  - Every file is associated with a specified group
  - Groupname restriction is the same as username restriction
    - Groupname can be the same as username
  - Each user has a default primary group and can belong to many group

User ID (*UID*) and Group ID (*GID*)

- Linux defines and tracks users and groups by numbers called UIDs and GIDs
- IDs from 0 to 99 is reversed for system use
  - UID=0 and GID=0 corresponds to ***root***
- Normal user can have UID and GID beyond 100 (or 500 in some distros)
  - Additional accounts will be associate the next-highest unused number
- *It's possible to create multiple account that use the same UID or GID*

Creating User Accounts

- Adding User: `useradd [options] username`

`-c comment`	comment for user
`-d home-dir`	account's home directory
`-e expire-date`	the date account will be disabled
`-g default-group`	name or GUID of the user's default group
`-G group[,…]`	names or GUIDs other groups user belongs to
`-m`	automatically create the user's home directory
`-k skeleton-dir`	specify another skeleton directory
`-M`	not automatically create the user's home directory
`-p encrypted-passwd`	passes the *pre-encrypted* password for the user
`-s shell`	user's default login shell
`-u UID`	specified user ID value
`-o`	allows user to have non-unique UID

Modifying User Account

- Setting user's password: **`passwd [options]`** **<u>`username`</u>**
  - **`-l`** locks an account be prefixing the encrypted password with ***!***
  - **`-u`** unlocks an account (by removing a leading ***!***)
  - **`-d`** removes the password from an account
  - **`-S`** display account's password information
- Modifying an existing account: **`usermod [options]`** **<u>`username`</u>**
  - **`usermod`** closely parallels **`useradd`** in its features and parameters
  - some additional options:
    - **`-m`** (when using with **`–d`**) move user's file to the new location
    - **`-l`** changes user's login name
    - **`-L`** locks a user's password
    - **`-U`** unlocks a user's password
  - if you change the UID, you should run **`chown`** to restores proper ownership on the files in user's home directory
    - Eg:

```
# usermod –u 510 sally
# chown –R sally /home/sally
```

Modifying User Account (cont')

- Modifying account expiration: **`chage [options] username`**
  - Linux accounts are automatically expire if either of two conditions is true
    1. The password hasn't been changed in a specified period of time
    2. The system date is past a predetermined time
  - Options:

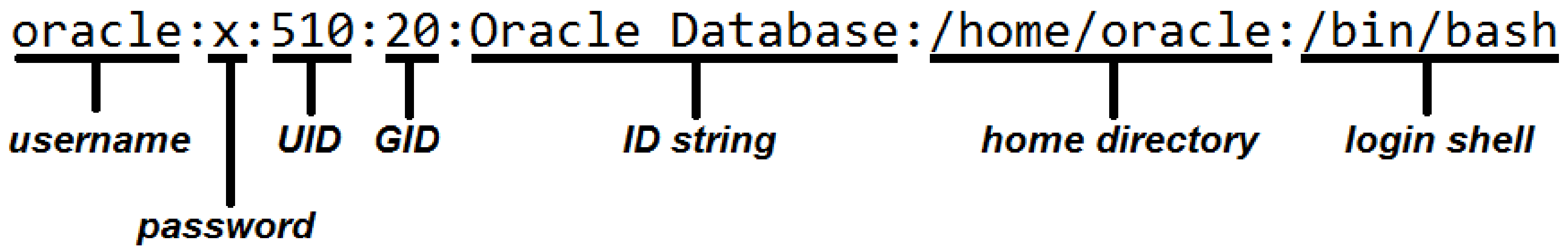
`-l`	display account expiration and password aging information
`-m mindays`	minimum number of days betwwen password changes
`-M maxdays`	maximum number of days between password changes
`-d lastday`	sets the last day a password was changed
`-I inactived`	sets the number of days between password expiration and account disablement
`-E expire`	set an absolute expiration date
`-W warndays`	number of days before account expiration that the system will warn the user

- Deleting accounts: **`userdel [options] username`**

`-r`	remove all files from the user's mail spool and home directory
`-f`	force deletion of the account while a user is logged in

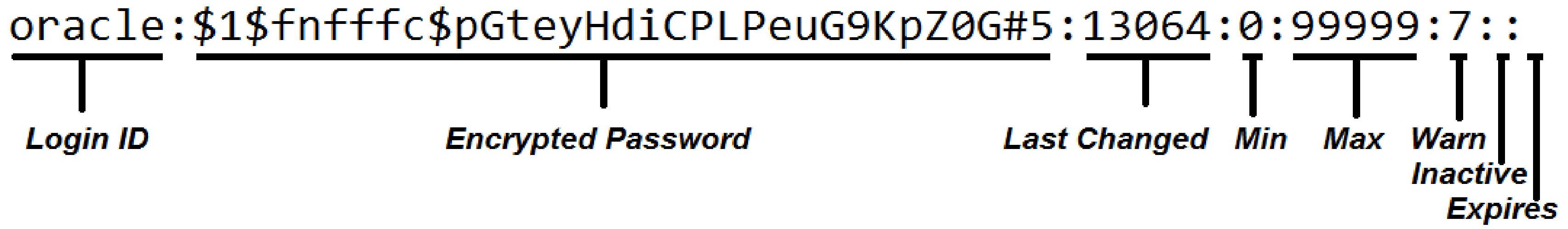
Directly Modifying User Configuration Files

- ***/etc/passwd***: containing basic information about users
  - password filed of ***x*** means system use a *shadow* password
  - **`vipw`** can be used for safety editing ***/etc/passwd***



```
oracle:x:510:20:Oracle Database:/home/oracle:/bin/bash
```

- ***/etc/shadow***: containing encrypted user passwords
  - passwords are encrypted using MD5 or DES (with *salt*)
  - date are stored as the number of day since *epoch* (1/1/1970)



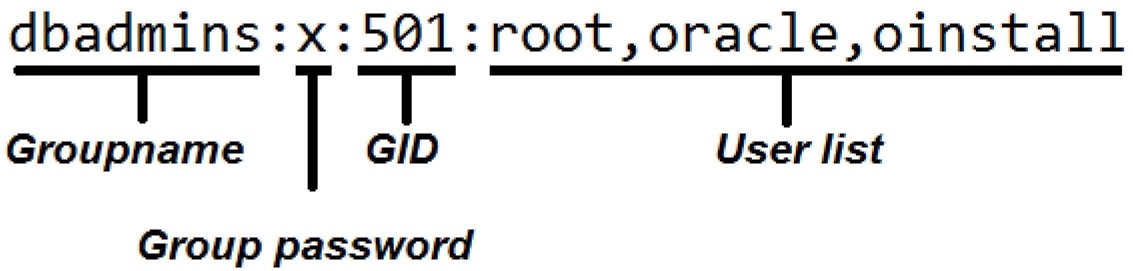
```
oracle:$1$fnfffc$pGteyHdiCPLPeuG9KpZ0G#5:13064:0:99999:7::
```

Configuring Group Account

- Adding Group: **`groupadd [options] groupname`**
  - **`-g GID`** specify GID
  - **`-o`** allows group to have a non-unique ID
  - **`-f`** return success even if the group already exist
  - **`-r`** (on some distros) create a group with GID of less than 500
- Modifies an existing group: ***groupmod [options] groupname***
  - **`-g GID`** specify new GID
  - **`-n newgroupname`** specify new groupname
- Add a user to groups: **`usermod –G group[,…] username`**
- Temporarily join a group: **`newgrp`**
- Set group password: **`gpasswd [options] groupname`**
  - **`-A user[,…]`** specify group administrators
  - **`-r`** removes the password from a group
- Deleting group: **`groupdel groupname`**

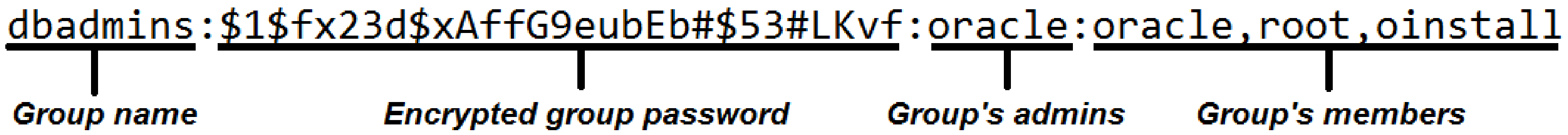
Directly Modifying Group Configuration Files

- ***/etc/group***: containing basic information about groups and wich users belong to them
  - **vigrp** can be used for safety edit ***/etc/group***



```
dbadmins:x:501:root,oracle,oinstall
```

- ***/etc/gshadow***: containing encrypted group passwords
  - passwords are encrypted using MD5 or DES (with *salt*)



```
dbadmins:$1$fx23d$xAffG9eubEb#$53#LKvf:oracle:oracle,root,oinstall
```

Exercise

1. Add a new group called ***group01*** with GID=1001
2. Add a new user called ***user01*** with UID=1002 and has a ***group01*** as the primary group
3. Set ***user01*** password to ***@bcd123$***
4. Try logging as ***user01*** to verify that the account works properly
5. View the permission mode for ***/etc/passwd, /etc/shadow, /etc/group*** and ***/etc/gshadow***
   - What’s the differences?
6. Use ***vipw*** to change the login shell of user01 to ***/bin/false***
7. Try to logging in as user01
   - Can ***user01*** log into the system?

2. Automate system administration tasks

Overview

Learn to:

- Manage ***cron*** and ***at*** jobs
- Configure user access to ***cron*** and ***at*** services

at and *cron* Services

- ***at***: executes commands at a specified time once
  - ***atd*** daemon checks its *job queue* every minute and executes the ones that are programmed to be run at that time
  - jobs are deleted from *job queue* after executed
- ***cron***: execute commands at a specified time repeatly
  - ***crond*** daemon wakes up every minute and checks the *crontabs* to determine what needs to be done

Using *at*

- Setting an ***at*** job: **`at [options] time`**
  - Options:

`-b`	run command only when the system load is low
`-d job#`	delete an at job from queue
`-f filename`	read job from a specified file
`-l`	list jobs for that user (all jobs for ***root***)
`-m`	mail user quen job completes
`-q queuename`	send the jobs to a queue (***a*** to ***z*** and ***A*** to ***Z***)

  - Time formats:

    now, 17:00, +3 hours, +2 minutes, +2 days, +3 months, 19:15 3.12.10, midnight, 4PM, 16:00 +3 days, mon, tomorrow …
- Show the at jobs queue of user: **`atq`** or **`at -l`**
- Deletes at jobs from the jobs queue: **`atrm job# [job#] …`**

Control Files for *at*

- ***/var/spool/atjobs***: where at jobs are stored
- ***/etc/at.allow***: list of users that are allowed to use at command
- ***/etc/at.deny***: list of users that are NOT allowed to use at command

at.allow	***at.deny***	**Users allowed to use *at***
NOT present	*NOT present*	only ***root***
NOT present	*present*	all users except the ones listed in ***at.deny***
present	*NOT present*	only users listed in ***at.allow***
present	*present*	only users listed in ***at.allow*** (***at.deny*** is ignored)

Examples with *at*

- Create an simple ***at*** job to run in 5 minutes later

```
$ at +5 minutes
echo “I am running in an at job”
[Ctrl-D]
```

- Create an ***at*** job which read commands from a file and run at midnight

```
$ at –f /tmp/myjob.sh midnight
```

- Using ***echo*** to run multiple commands with ***at***

```
$ echo ‘cd /tmp; ls –a’ | at 12:00 tue
```

- Listing all ***at*** jobs

```
$ at –l
$ atq
```

Using *cron*

- *Crontabs* are configuration files read by the ***crond*** daemon defining which jobs should be run when
- Users crontabs: created by **`crontab -e`**
  - Saved in ***/var/spool/cron/crontab/<username>***
  - Format of the user crontab: 6 filelds per line

```
#minute  hour  dayofmonth  month    dayofweek    command
#0-59    0-23  1-31        1-12     0-7(1=Mon)   <valid command>
 0       21    15          3,6,9    *            /opt/report.sh
```

- System wide crontabs: listed in ***/etc/crontab*** and ***/etc/cron.d*** directories
  - Format of the system crontab: 7 filelds per line

```
#minute   hour  dayofmonth   month dayofweek [username]   command
#0-59     0-23  1-31         1-12  0-7        user        <valid command>
02        4     *            *     *          root        run-parts /etc/cron.daily
```

crontab commands

- Create/edit a user crontab: **`crontab –e`**
- Create a user crontab by reading from file: **`crontab –e filename`**
- Display user's crontab file: **`crontab –l`**
- Delete user's crontab file: **`crontab –r`**
- Edit a user's crontab file (for ***root*** only): **`crontab –e –u username`**

Access control of *cron*

- ***/etc/cron.allow***: list of users that are allowed to use ***cron***
- ***/etc/cron.deny***: list of users that are NOT allowed to use ***cron***

cron.allow	***cron.deny***	**Users allowed to use *cron***
NOT present	*NOT present*	any users
NOT present	*present*	all users except the ones listed in ***cron.deny***
present	*NOT present*	only users listed in ***cron.allow***
present	*present*	only users listed in ***cron.allow*** (***cron.deny*** is ignored)

Examples of user crontabs

- Run a command every hour from 8:00 to 18:00 everyday

```
0   8-18  *     *     *     <command>
```

- Run a command every 4 hours on the half hour (i.e *6:30, 10:30, 14:30, 16:30*) everyday

```
30  6-16/4      *     *     *     <command>
```

- Run a command every day, Monday to Friday at 01:00, and doesn't report to syslog

```
-0  1     *     *     1-5   <command>
```

- Run the command every Monday and Tuesday at *12:00, 12:10, 12:20, 12:30*

```
0,10,20,30    12    *     *     1,2   <command>
```

- Run a command every 10 minutes

```
*/10  *     *     *     *     <command>
```

Another Scheduling Services

- ***batch***: similar to ***at***, except that jobs are run only when the system load is low (system load average < 0.8)
- ***anacron***: similar to ***cron***, but only handle scheduling of the jobs usually done *daily*, *weekly* or *monthly* (not *hourly* jobs)
  - jobs are listed in ***/etc/anacrontab***

```
#period    delay     job-identifier   command
1          5         dailyrun         run-parts /etc/cron.daily
7          10        weeklyrun        run-parts /etc/cron.weekly
30         15        monthlyrun       run-parts /etc/cron.monthly
```

Exercise

1. Create a system crontab to delete all files ending with ***.log*** in the ***/tmp*** directory at 9PM every Sunday
2. Create a user crontab to report the usage of user’s home directory every hours, from 8:00AM to 17:00PM on all working days (Monday to Friday).

Hints to Exercise

1. `echo “00 21 * * 7 root rm -f /tmp/*.log” >> /etc/crontab`
2. `crontab -e`

```
00 8-17 * * 1-5 du -sh $HOME >> /tmp/diskreport
```

3. Localisation and internationalisation

Overview

Learn to:

- Locale settings
- Timezone settings

Setting Time Zone

- Linux uses UTC internally
- ***/etc/localtime*** contains information about local timezone
  - not a plain-text file
  - can be a symbolic link to file in ***/usr/share/zoneinfo***
- Steps to change timezone:
  1. Find your timezone file in ***/usr/share/zoneinfo***
     - Example: **`/usr/share/zoneinfo/US/Eastern`**
  2. Remove or delete the old ***/etc/localtime*** file
     - Example: **`rm /etc/localtime`**
  3. Create a symbolic link from your chosen timezone file to the ***/etc/localtime*** file
     - Example: **`ln -s /usr/share/zoneinfo/US/Eastern /etc/localtime`**
- GUI tools to change timezone: **`tzsetup`**, **`tzselect`**, **`tzconfig`**

What is a Locale

- A way of specifying the computer’s language, country and related information for customizing displays
- Locale form: `[language[_territory][.codeset][@modifier]]`
  - **`language`**: codes for language (**`en`**, **`fr`**, **`ja`**…)
  - **`territory`**: codes for nations (**`US`**, **`FR`**, **`JP`**…)
  - **`codeset`**: encoding names (**`ASCII`**, **`ISO-8859`**, **`UTF-8`**…)
  - **`modifier`**: locale-specific code that modifies how it works (Eg: sort order)
  - Example: **`en_GB.UTF-8`**

What Is Your Locale

- View your current locale: **`locale`**

```
LANG=“en_US.UTF-8”
LC_NUMERIC=“en_US.UTF-8”
LC_TIME=“en_US.UTF-8”
LC_MONETARY=“en_US.UTF-8”
LC_PAPER=“en_US.UTF-8”
LC_ADDRESS=“en_US.UTF-8”
LC_ALL=“en_US.UTF-8”
```

- **`LC_ALL`** can overrides all other **`LC_*`** value
- **`LANG`** sets the locale in case the **`LC_*`** aren’t set
  - **`LANG=C`**: programs will display output without passing it through locale translations
    - Helps to avoid some types of problems in pipelines and scripts

Changing Your Locale

- View all the available locales: **`locale -a`**
  - You may install additional locale packages
- Temporarily change your locale:

```
export LC_ALL=<locale_code>
export LANG=<locale_code>
```

- Permanently change your locale: Adjust bash startup script files (***~/.bashrc, /etc/profile***)
- Modifying Text-File Locales:

```
iconv –f encoding –t encoding [inputfile] …
```

  - Eg: `iconv –f iso-8859 –t UTF-8 file1.txt > file2.txt`

Exercise

1. View your current locale settings
2. Run **`date`** command to see the format of command's output according to current locale setting
3. View all your available locale settings
4. Temporarily change your current locale settings to another locale code (example: **`vi_VN.utf8`**)
5. Run **`date`** command again to see the new output format

Thank You !

BACKUP SLIDES