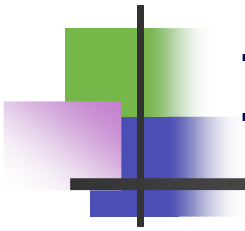


107. Tâches d’administration

Sujet 107 :Tâches d'administration

- 107.1 Gestion des utilisateurs, des groupes et des fichiers associés (Weight5)
- 107.2 Automatisation de tâches d’administration par la planification de lancement de programmes (Weight 4)
- 107.3 Localisation et internationalisation. (Weight 3)

Gestion des utilisateurs, des groupes et des fichiers associés

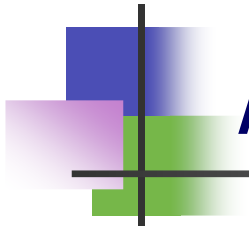
- **Description** : Les candidats doivent être capables d'ajouter, de détruire, de suspendre et de modifier des comptes d'utilisateurs.
- **Termes, fichiers et utilitaires utilisés :**
  - /etc/passwd, /etc/shadow , /etc/group, /etc/gshadow.
  - chage, gpasswd, groupadd, groupdel, groupmod, passwd, useradd, userdel, usermod

Base de données des utilisateurs :/etc/passwd

- Format :
  account:passwd:UID:GID:GECOS:directory:shell
  - **identifiant** (ou **login**), par exemple salah;
  - **mot de passe :** il s'agit d'un mot de passe chiffré e par la fonction a sens unique crypt ou md5. La valeur spéeciale < x > indique que le mot de passe chiffré est stocké dans **/etc/shadow ;**
  - **uid** : numéro unique identifiant l'utilisateur ;
  - **gid** : numéro unique du groupe principal de l'utilisateur (Debian crée par défaut un groupe spécifique à chacun) ;

Base de données des utilisateurs :/etc/passwd

- **GECOS** : champ de renseignements qui contient habituellement le nom complet de l'utilisateur ;
- **répertoire de connexion,** attribué à l'utilisateur pour qu'il y stocke ses fichiers personnels (la variable d'environnement **$HOME** y pointe habituellement) ;
- **Programme à exécuter après la connexion**. Il s'agit généralement d'un interpréteur de commandes (shell). Si l'on précise **/bin/false** (programme qui ne fait rien et rend la main immédiatement), l'utilisateur ne pourra pas se connecter.



/etc/shadow

- Fichier de mots de passe chiffrés.
- Format (cf shadow (5))
  - **Identifiant ou login**
  - **Mot de passe chiffré**
  - **Plusieurs champs de gestion de l'expiration du mot de passe.**
- Améliorer la sécurité : stocker les mots de passe chiffrés dans **/etc/shadow**.
  - Créer **/etc/shadow** à partir de **/etc/passwd** : **pwconv**
  - Stocker les mots de passe dans **/etc/passwd** :**pwunconv**

/etc/group

- Base de données des groupes
- **Un groupe Unix** est une entité regroupant plusieurs utilisateurs afin qu'ils puissent facilement se partager des fichiers à l'aide du système de droits intégré (en jouissant justement des mêmes droits).
- Format (ch group (5))
  - **identifant** (le nom du groupe) ;
  - **mot de passe (facultatif)**
  - **gid** : numéro unique identifiant le groupe ;
  - **liste des membres** : liste des utilisateurs membres du groupe, séparée par des virgules.

Création de compte

#useradd -m -g auf -c "Mohamed Salah" -s /bin/tcsh msalah

passwd msalah

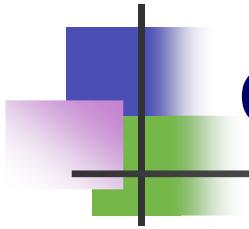
- **/etc/default/useradd** (SUSE) ou **/etc/adduser.conf** (debian) offre quelques paramétrages par défaut au compte créé avec **useradd** :
  - GROUP=100 HOME=/home INACTIVE=-1 EXPIRE= SHELL=/bin/bash SKEL=/etc/skel GROUPS=video,dialout CREATE_MAIL_SPOOL=no

Création de compte

- **useradd -D :** Paramètres par défaut de la commande useradd
- **useradd** fabrique le répertoire personnel et y recopie le contenu du répertoire modèle **/etc/skel**.

Gestion des groupes

- groupadd auf : créer un nouveau groupe auf
- groupdel auf : supprimer le groupe
- groupmod : modifier les informations d'un groupe
  - groupmod -n AUF auf



Gérer les comptes

- **passwd** : changer le mot de passe.
- **chfn** (Change Full Name) : intervient sur le champs GECOS
- **chsh** (Change Shell) : permet de changer le shell de login (le choix du shell est limité à la liste dans **/etc/shells)**
- **userdel -r utilisateur** : supprimer utilisateur ainsi que le répertoire personnel et les fichiers de l'utilisateur.

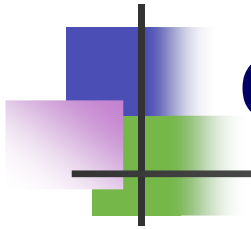
- **usermod** : Modifier un compte
  - Changer le groupe primaire **:** usermod -g users msalah
  - Changer les groupes secondaires **:** usermod -G stagiaire,prof msalah

- Bloquer un compte d'un utilisateur : l'empêcher de se connecter à nouveau :

Bloquer	**Débloquer**
passwd -l	**passwd -u**
usermod -L	**usermod -U**

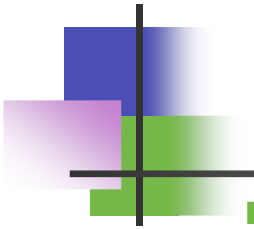
/etc/login.defs

- Les caractéristiques par défaut du mot de passe sont définis à partir du fichier **/etc/login.defs**, sauvegardés dans le fichier ***/etc/shadow***
  - Bagdad:~ # grep -v '^#' /etc/login.defs
  - PASS_MAX_DAYS 99999
  - PASS_MIN_DAYS 0
  - PASS_WARN_AGE 7
  - SYSTEM_UID_MIN 100
  - SYSTEM_UID_MAX 499
  - UID_MIN 1000
  - UID_MAX 60000
  - SYSTEM_GID_MIN 100
  - SYSTEM_GID_MAX 499
  - GID_MIN 1000
  - GID_MAX 60000



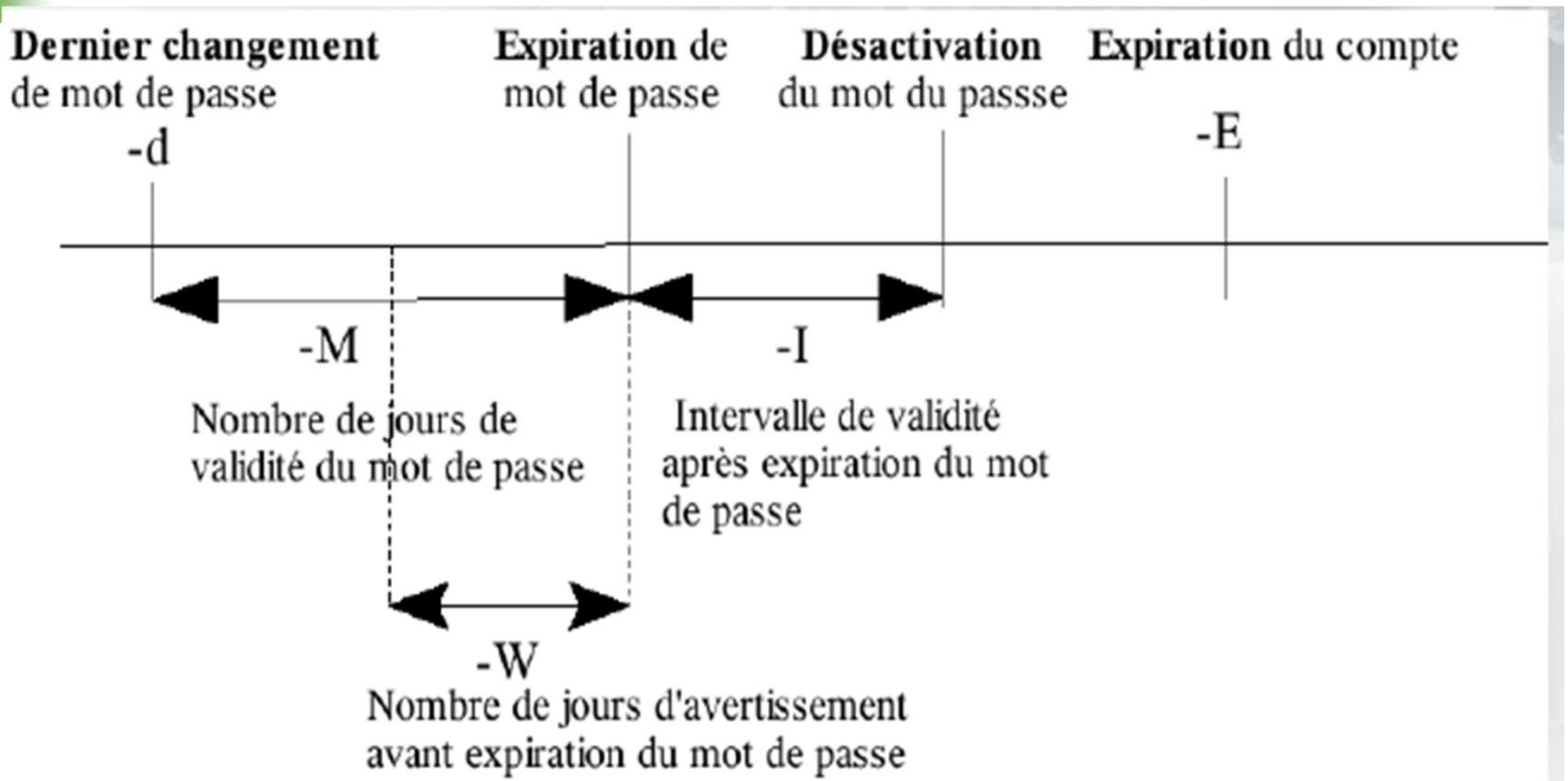
chage

- La commande **chage** permet de modifier ces informations contenues dans le fichier **/etc/shadow**



- chage [-m mindays] [-M maxdays] [-d lastday] [-I inactive] [-E expiredate] [-W warn days] user
- chage -l [user]
  - **l** donne les valeurs actuelles du compte
  - **E** permet de fixer une date d'expiration sous la forme Unix standard (nombre de jours depuis le 1er janvier 1970) ou sous la forme YYYY/MM/DD
  - **M** permet de changer la valeur du nombre maximal de jours de validité du mot de passe.
  - **-m** permet de changer la valeur du nombre minimal de jours de validité de mot de passe.
  - permet de changer le nombre de jours d'alerte avant un changement obligatoire de mot de passe.
  - **-d** permet de changer la date de dernier changement de mot de passe sous forme Unix standard

Chage (suite)


Dernier changement de mot de passe
Expiration de mot de passe
Désactivation du mot du passse
Expiration du compte
-d
-E
-M
-I
Nombre de jours de validité du mot de passe
Intervalle de validité après expiration du mot de passe
-W
Nombre de jours d'avertissement avant expiration du mot de passe

Automatisation de tâches d'administration

- **Description** : Les candidats doivent être capables d'utiliser les commandes "**cron**" et "**anacron**" pour lancer des programmes à intervalles réguliers et d'utiliser la commande "**at**" pour lancer des programmes à des moments spécifiques
- **Termes, fichiers et utilitaires utilisés :**
  - /etc/anacrontab /etc/at.deny /etc/at.allow /etc/crontab /etc/cron.allow /etc/cron.deny /var/spool/cron/*

    at atq atrm crontab.

Planification synchrone : cron et atd

- **cron** : démon chargé d'exécuter des commandes planifiées et récurrentes (chaque jour, chaque semaine, etc.) ;
- **atd** : démon qui s'occupe des commandes à exécuter une seule fois à un instant précis, à un instant précis et futur
- De nombreuses tâches sont régulièrement planifiées
  - la rotation des logs;
  - mise à jour de la base de données du programme locate;
  - les sauvegardes;
  - des scripts d'entretien (comme le nettoyage des fichiers temporaires).

cron

- Tous les utilisateurs peuvent planifier l'exécution de tâches. c'est pourquoi chacun dispose de sa propre **crontab**.
- Un utilistaeur peut éditer les commandes à planifier :
  - **crontab –e**
  - ses informations sont stockées dans **/var/spool/cron/crontabs/<utilisateur>**).
- Afficher la liste des tâches programmées
  - **crontab -u salah –l**
  - **35 * * * * echo "hello" | mail zied@auf.com**
  - **00 * * * * /usr/X11R6/bin/xclock –display :0.0**
- Supprimer la table cron
  - **crontab -u salah –r**

Format d'un fichier crontab

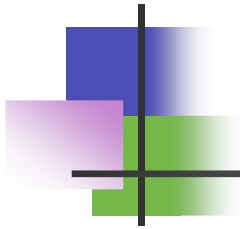
- Chaque ligne significative d'une **crontab** décrit une commande planifiée grâce aux six champs suivants :
  - la condition sur les **minutes** (nombres compris de 0 a 59) ;
  - la condition sur les **heures** (de 0 a 23) ;
  - la condition sur le **jour du mois** (de 1 a 31) ;
  - la condition sur le **mois** (de 1 a 12) ;
  - la condition sur le **jour de la semaine** (de 0 a 7, 0 et 7 correspondant au dimanche ; il est également possible d'employer les trois premières lettres du nom du jour en anglais comme Sun, Mon, etc.) ;
  - **la commande a exécute**r (quand toutes les conditions précédentes sont remplies).

Format d'un fichier crontab

- La syntaxe ***a-b*** décrit l'intervalle de toutes les valeurs comprises entre a et b
- La syntaxe **a-b/c** décrit un intervalle avec un incrément de c (exemple : 0-10/2 correspond a 0,2,4,6,8,10.

Exemple de crontab

- #Format
- #min heu jou moi jsem commande
- # Télécharge les données tous les soirs à 19:25
  25 19 * * * $HOME/bin/get.pl
- # Le matin a 8h00, en semaine (lundi à vendredi)
  00 08 * * 1-5 $HOME/bin/fait_quelquechose
- # Redémarre le proxy IRC après chaque reboot
  @reboot /usr/bin/dircproxy
- # Au début de chaque heure
- 00 * * * * $HOME/bin/fait_une_autrechose.pl



at

- La commande **at** prévoit l'exécution d'une commande à un moment ultérieur.
  - **Exemple 1**: Planifier une tâche à 1:23 am

    $ at 1:23am

    at>lp /usr/sales/reports/*

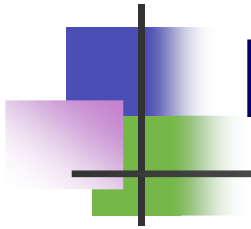
    at> echo "Files printed, Boss!" | mail -s"Job done" boss

    at> ^D
  - Exemple 2 :Exécuter les commandes listées dans le fichier command_list après deux jours à 9 pm

    $ at -f command_list 9pm + 2 days

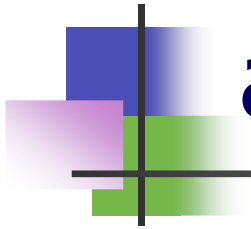
Sécurité : Restreindre l'accès à cron et atd

- **cron.allow, cron.deny**
- **at.allow, at.deny**
- **Si le fichier « allow » existe** : SEULS les usagers listés dans ce fichiers sont autorisés à utiliser le service.
- **Si le fichier « allow » n'existe pas, mais le fichier « deny » existe**, SEULS les usagers non listés dans le fichier « deny » sont autorisés à utiliser le service.
- **cron** : si les deux fichiers **cron.allow** et **cron.deny** n'existent pas, alors tous les usagers peuvent utiliser le service.
- **atd** : si les deux fichiers **at.allow** et **at.deny** n'existent pas, seul **root** peut utiliser le service.
- **atd** :un fichier **at.deny** vide, donne l'accès à **atd** à tous les usagers. **(configuration par défaut)**



Exercices

- 1. Créer une entré cron qui permet de lancer xclock chaque deux minutes.
- 2. Utiliser at afin de lancer xclock dans les deux minutes suivantes.



at

- Lister les commandes actuellement planifiés : **atq ou at -l**
  - $ at –l

  12 2009-05-28 18:15 a salah
- Annuler une commande planifiée : **atrm , at -r ou at –d**
  - $ atrm 12