# TP2-UTC502

## Gestion des utilisateurs


**Groupe 1 :**

BIGAUD Emilie

DOUÇOT Kévin

LE COZ Julien

PARIS Loic

SARAHANABHAVAN Saranya

## Manipulation 1 : Manipulation des fichiers

Méthode dite **naïve** par l'édition des fichiers */etc/passwd* */etc/shadow* et /*etc/group*

- **Ajout de 2 groupes**

Création des groupes en utilisant la commande *vi*, se positionner dans */etc/group*

Les informations concernant les groupes sont conservées dans le fichier */etc/group* et contient 4 champs :

| **Champs** | **Contenu** |
|---|---|
| Gname | Nom du groupe |
| Mot de passe | Rarement utilisé |
| GID | Identificateur du groupe |
| Liste des membres | Liste des logins utilisateurs membre du groupe. Les utilisateurs dont c'est le groupe principal n'ont pas besoin d'apparaitre |

```
#vi /etc/group
```

Que ce soit pour l'UID ou GID, les valeurs comprises entre 0 et 99 sont réservés à des comptes système. A partir de 100 cela peut être des comptes utilisateurs. Par défaut lors d'une création avec les commandes *useradd* et *groupadd* la valeur commence à 1000. Nous avons choisi arbitrairement des identificateurs à 1200, 1201 et ajouté des deux lignes à la fin du fichier :

```
auditeurs:x:1200:
enseignants:x:1201:
:wq                    #sauvegarde et fermeture du fichier
```

- **Ajout de l'utilisateur « *tintin* »**

Rappel des champs pour les fichiers */etc/passwd*

| Champs | Contenu |
|---|---|
| Uname | Login |
| Mot de passe | Le mot de passe est crypté. Quand le système shadow est utilisé, ce champ contient le caractère x |
| UID | Numéro d'identification de l'utilisateur |
| GID | Numéro d'identification du groupe d'appartenance de l'utilisateur |
| Commentaire | A titre d'information, par exemple Prénom/Nom de l'utilisateur |
| Répertoire de connexion | Répertoire personnel de l'utilisateur |
| Commande de connexion | Shell initial exécuté à la connexion |

Rappel des champs pour le fichier */etc/shadow*

| Champs | Contenu |
|---|---|
| Login | Login |
| Mot de passe | Une * dans ce champ indique le compte ne peut être connecté (cas du compte bin par exemple). Si commençant par *!!* indique que le compte est verrouillé. |
| Age | Le nombre de jour écoulé depuis le 1er janvier 1970 et la date de mise à jour du mot de passe. |
| Période de changement | Le nombre minimum de jours entre deux changements de mots de passe. Un 0 indique que l'utilisateur peut changer le mot de passe à n'importe quel moment. |
| Durée de validité | Le nombre maximum de jours pendant lesquels le mot de passe est valide. La valeur 99999 indique que le mot de passe est toujours valide. |
| Durée de validité restant | Nombre de jours avant l'expiration |
| Durée d'invalidité | Nombre de jour après l'expiration provoquant la désactivation du compte. Un champ vide indique qu'il n'y a aucune désactivation |
| Date d'expiration | Exprimée en nombre de jour depuis la date de référence |
| Champs réservé | |

1. Ajout d'une ligne pour l'utilisateur

```
# vi /etc/passwd
tintin:x:1001:1001::/home/tintin:/usr/bin/zsh # Choix arbitraire de l'UID 1001 pour "tintin"
:wq
```

```
vi /etc/shadow
tintin:tintin:19331:0:99999:7::: #mot de passe en clair car pas moyen de le crypter sans "passwd"

:wq
```

- 1er test de connexion

```
# su tintin
tintin@kali:/$
```

Il n'y a pas d'erreur et ne demande pas de mot de passe. En effet, aucun n'a été définit et crypté. Lorsque nous essayons de nous connecter, le message est :

```
Your password is incorrect. Please try again.
```

## 2. Positionnement du mot de passe initial

Utilisation de la commande *passwd* :

```
# passwd tintin
New password :
Retype new password:
passwd: password updated successfully
```

Contrôle du fichier shadow :

```
tintin:$y$j9.......Mn6:19449:0:99999:7:::
```

Nous pouvons observer que le mot de passe a été modifié et crypté.

- 2ème test de connexion

Lorsque nous essayons de nous connecter, plus de message d'erreur de mot de passe, mais il nous est impossible d'arriver jusqu'au bureau.

## 3. Création du répertoire Personnel de l'utilisateur

Utilisation de la commande *mkdir* :

```
# mkdir /home/tintin
```

Vérification de la création du dossier :

```
ls -l /home/
drwx------ 15 kali kali 4096 Apr 1 13:09 kali
drwxr-xr-x 2 root root 4096 Apr 3 00:12 tintin
```

Nous voyons le dossier créé, mais le propriétaire est root.

- 3ème test de connexion

Aucun changement par rapport à l'étape 2

4. Changement de propriétaire pour ce répertoire

Utilisation de la commande *chmod* pour modifier les droits d'accès pour le propriétaire uniquement :

```
# chmod -R 700 /home/tintin
ls -l /home/
drwx------ 2 root root 4096 Apr 3 00:12 tintin
```

Utilisation de la commande *chown* pour le changement de propriétaire :

```
# chown -R tintin /home/tintin
ls -l /home/drwx------ 2 tintin root 4096 Apr 3 00:12 tintin
```

- 4ème test de connexion

Il est possible d'ouvrir une session *tintin*.

5. Copie des fichiers d'environnement

Utilisation de la commande *cp* pour copier le dossier.

```
# cp -r /etc/skel/ /home/tintin/
```

La copie c'est correctement déroulé, mais il n'y avait aucun fichier dans le dossier */etc/skel/*

- 5ème test de connexion

Il est possible d'ouvrir une session avec l'utilisateur *tintin*.

## Synthèse

La méthode "naïve" fonctionne, mais cela est assez fastidieux du fait de sa longueur de mise en place. Il y a un risque non négligeable de commettre des erreurs et de ce fait créer des connexions non voulues voir même non sécurisé pour l'entreprise ou empêcher le bon fonctionnement des services pour un ou des utilisateurs.

## Manipulation 2 : Commandes de base

| **Commande** | **Action** |
|---|---|
| Adduser | Ajout d'un utilisateur de façon complet |
| Useradd | Ajout d'un utilisateur simple |
| Usermod | Modifie les paramètres d'un utilisateur (login, accès) |
| Userdel | Suppression d'un utilisateur |
| Groupadd | Ajout d'un groupe |
| Groupmod | Modifie les paramètres du groupe (login, accès) |
| Groupdel | Suppression d'un groupe |
| Pwck | Vérifie l'intégralité des fichiers MDP utilisateur (*/etc/passwd* et */etc/shadow*) |
| Grpck | Vérifie l'intégralité des fichiers administration des groupes (*/etc/group* et */etc/gshadow*) |
| Finger | Affiche les informations de tous les utilisateurs, des options permettent d'avoir plus de détail |
| Chfn | Change nom utilisateur et les informations le concernant |
| Chsh | Change le login du shell |
| Passwd | Définir et modification d'un mot de passe utilisateur |
| Su | Permet d'effectuer une action en tant qu'utilisateur en particulier |
| Groups | Indique le login appartenant au(x) groupe(s) de façon simple |
| id | Comme *Groups*, en indiquant la valeur des GID et UID |
| Vipw | Modifie le fichier des MDP */etc/passwd* avec l'option -s il y a la version sécurisé */etc/shadow* |
| Vigr | Modifie le fichier des groupes */etc/group* avec l'option -s il y a la version sécurisé */etc/gshadow* |

La commande *adduser* est de plus haut niveau que *useradd* car cela permet de créer un profil complet, avec notamment la création d'un répertoire personnel, d'un mot de passe et l'inclusion dans un groupe. Lors de la création, la personne pourra utiliser son compte immédiatement. Tandis que *useradd* est plus basique, cela créé un compte simple mais il faudra lancer d'autres commandes, comme par-exemple *passwd* pour attribuer un mot de passe. Cette commande est utile si on souhaite créer un compte temporaire le temps d'une journée.

## Manipulation 3 : Collecte des informations

### 1. Utilisateur *bin*

Pour vérifier la présence de l'utilisateur « *bin* » nous utilisons la commande *grep*

```
# grep ^bin /etc/passwd          #Le ^ permet de cherche les caractères en début de ligne
bin:x:2:2:bin:/bin:/usr/sbin/nologin
```

L'utilisateur "*bin*" existe avec un UID et GID à 2.

### 2. Comptes similaires aux mêmes droits que *root*

En cherchant dans le fichier */etc/group/*, nous pouvons vérifier s'il y a d'autres utilisateurs appartenant au groupe *root*.

```
# grep root /etc/group
root:x:0:
```

Il n'y a pas d'autres utilisateurs. Ceci reste cohérent car l'utilisateur *root* est unique et dispose de tous les droits. Il est le seul à avoir un UID spécial à 0.

Cependant, d'autres utilisateurs peuvent avoir des droits privilégier, ils sont nommés les super-utilisateurs, la commande *sudo* permet des droits supplémentaires. Sous la VM Kali c'est le cas :

```
grep sudo /etc/group
sudo:x:27:kali                #kali est renseigné comme super-utilisateur
```

### 3. Groupe de *bin*

Nous avons vu que le GID est à 2. Pour s'en assurer et comme vu lors de la manipulation 2, la commande id permet d'avoir les informations sur l'identité d'un compte

```
# id bin
uid=2(bin) gid=2(bin) groups=2(bin)
```

Ou encore avec la commande *groups*

```
# groups bin
bin:bin
```

## Synthèse

L'utilisateur *bin* est un utilisateur système unique qui joue un rôle bien précis : exécuter des programmes binaires nécessaires au fonctionnement du système, comme par-exemple l'installation de nouveaux paquets ou la gestion des systèmes de fichiers.

Il est souvent utilisé pour de l'installation ou de la configuration. Ces tâches nécessitent des privilèges élevés, mais ne nécessitant pas les accès complets comme *root*. L'utilisateur *bin* est un super-utilisateur disposant de permission particulière, en n'ayant pas le niveau d'accès complet et illimité comme l'utilisateur *root* qui est unique.

## Manipulation 4 : Création et gestion d'un utilisateur

### 1. Création de l'utilisateur

Se positionner en *sudo su* pour passer la commande *useradd*

```
# useradd pierre
```
#Comme vu précédemment, cela créer simplement l'utilisateur

Les paramètres par défaut de cette commande peuvent être modifié à cet emplacement */etc/default/useradd*

Vérification du groupe de pierre :

```
# groups pierre
pierre : pierre
```

Par défaut, il appartient à son propre groupe car nous n'avons pas précisé dans lequel il doit se trouver.

Vérification du Shell de pierre :

```
# grep "pierre" /etc/passwd
pierre:x:1002:1002 :: /home/pierre:/bin/bash
```

Le shell de l'utilisateur pierre est : */bin/bash*. Nous pouvons également voir que son répertoire est */home/pierre*.

### 2. Ajout d'un groupe

Passer la commande *groupadd* :

```
# groupadd staff
groupadd: group 'staff' already exists
```

Ajout de pierre dans le groupe staff avec la commande *usermod* :

```
# usermod -aG staff pierre
```
#valeur a : Ajout à un groupe secondaire
#valeur G : Ajout groupe supplémentaire

Affichage des groupes de pierre avec la commande *groups* :

```
# groups pierre
pierre : pierre staff
```

Pierre appartient désormais aux groupes ***pierre*** et ***staff***

## 3. Connexion aux comptes

Avec une connexion classique, il est impossible de se connecter car aucun mot de passe n'a été définis. Il manque également dans le dossier */home/pierre* les fichiers du répertoire */etc/skel*.

Pour commencer, création du mot de passe avec la commande *passwd* :

```
# passwd pierre
New password :
Retype new password:
passwd: password update successfully
```

Création du dossier */home/pierre/* et modification des droits avec la commande *chmod* :

```
# mkdir /home/pierre/
chmod -R 700 /home/pierre/
```

Copie des fichiers du répertoire */etc/skel* dans */home/pierre/* :

```
# cp -r /etc/skel/ /home/pierre/
```

# Valeur r : pour récursive, tous les fichiers inclus dans le dossier copié

Modification du propriétaire :

```
# chown -R pierre:pierre /home/pierre/
```

Contrôle du fichier copié et modifié :

```
# ls -l /home/
total 16
drwx------ 15 kali kali 4096 Apr 3 22:36 kali
drwx------ 16 pierre pierre 4096 Apr 3 23:11 pierre
```

La connexion classique est maintenant possible à la suite des commandes passé précédemment.

La commande *su* permet de se connecter en tant qu'utilisateur "Pierre" depuis un autre compte sur un terminal. Tandis que la connexion classique permet de se connecter via une interface graphique.

## 4. Changement du champ commentaire

Pour changer le champ commentaire nous utilisons la commande *vipw* :

```
# grep "pierre" /etc/passwd
pierre:x:1002:1002:Pierre Cerf-Lannion:/home/pierre:/bin/sh
```

## 5. Verrouiller et déverrouiller un compte

- Verrouillage du compte

Utilisation de la commande *usermod* :

```
# usermod -L pierre
```

Cette commande ajoute un **!** devant le **$** du mot de passe crypté dans le fichier */etc/shadow*, il est donc possible de le faire avec l'éditeur en *vi* :

```
# grep "pierre" /etc/shadow
pierre:!$y$j9T$N.OVAioZDUzUaHgLH2ANN.$a/US054kWiwc4tZw7S5l/VwrF9veqWQY4.wN3
uToHtC:19450:0:99999:7:::
```

Pour vérifier l'état d'un utilisateur, il est également possible d'utiliser la commande suivante :

```
su pierre
Password:_
su: Authentication failure
```

Utilisation de la commande *passwd* :

```
# passwd -l pierre          #Verrouillage session

# passwd -S pierre
pierre L 2023-04-03 0 99999 7 -1
```

La valeur **L** indique que le compte est *Locked*, ou verrouillé.

- Déverrouillage du compte

```
# usermod -U pierre
```

Vérification du fichier /*etc/shadow* :

```
# grep "pierre" /etc/shadow
pierre:$y$j9T$N.OVAioZDUzUaHgLH2ANN.$a/US054kWiwc4tZw7S5l/VwrF9veqWQY4.wN3u
ToHtC:19450:0:99999:7:::
```

Vérification avec commande *passwd* :

```
# passwd -u pierre          #Dévérouillage session

# passwd -S pierre
pierre P 2023-04-03 0 99999 7 -1
```

La valeur P indique que la session de l'utilisateur est déverrouillée.

La connexion en interface de commande ou interface graphique fonctionne correctement.

### 6. Création compte avec outil d'administration

A la création du compte *admin* il est **impossible** de choisir un UID 0. Le message est le suivant :

```
User ID 0 is already used by user root. Please choose a different numeric identifier for admin.
```

Nous ne pouvons pas choisir un UID identique à "root". Il est possible par-contre de l'intégrer au groupe *sudo* et donc d'avoir les droits de *root* en utilisant la commande *sudo* avant chaque commande.

## Synthèse

Parmi les méthodes permettant de créer un utilisateur, il est possible de faire la configuration soi-même grâce à *useradd*, d'utiliser une commande de plus haut niveau telle que *adduser* ou encore d'utiliser un programme en interface graphique. Quoi qu'il en soit certaines règles ne peuvent être transgressé. Le compte administrateur *root* est bien spécifique et il n'est pas possible de lui créer un alias en attribuant son UID à un autre profil. Cependant il est possible d'accorder les privilèges d'administrateur à un utilisateur classique. Ainsi on s'assure que l'administrateur *root* garde les pleins pouvoirs sur le système et soit toujours en mesure de corriger d'éventuels dysfonctionnements ou problèmes de configuration.

## Manipulation 5 : Gestion des utilisateurs suite

1. Que fait la commande ID

La commande *id* permet d'afficher pour un utilisateur donné (passé en paramètre, par défaut s'il n'y a pas de paramètre l'utilisateur courant) son UID, son GID et ses groupes. Par-exemple :

```
uid=1001(admin) gid=1001(admin) groupes=1001(admin),100(users)
```

2. Etapes pour fermer un compte utilisateur

- Sauvegarde des données

```
cp -r /home/user_to_delete /home/save_user_to_delete
```
# r permet de copier des arborescence en récursif

Une autre option est également possible, le *--backup* ou *-b* qui permet, si un fichier au identiques existe, de rajouté un caractère pour le différencié, un ~ par exemple.

```
cp --backup /home/user_to_delete /home/save_user_to_delete
```

- Suppression du compte

```
userdel -r user_to_delete
```

### Synthèse

Il est bien sûr possible de supprimer un utilisateur. Cependant cet utilisateur dispose d'un dossier courant qui est l'endroit privilégié pour entreposer ses documents, documents qu'il peut être nécessaire de garder à des fins d'archives par-exemple. Il existe donc la possibilité de supprimer l'utilisateur avec le dossier correspondant ou sans.

## Manipulation 6 : Gestion des groupes

La commande *newgrp* permet de changer l'identifiant de groupe de l'utilisateur au cours d'une session. Si l'option - est fournie, l'environnement de l'utilisateur est réinitialisé, comme si l'utilisateur venait de se connecter. Sinon, l'environnement actuel, y compris le répertoire de travail actuel est conservé.

```
$ Su pierre
# groups
pierre staff
```

Utilisation de la commande *newgrp*

```
# newgrp staff
# groups
staff pierre
```

Dans tous les cas, l'utilisateur appartient à tous les groupes affichés, ses droits sur ces groupes sont inchangés.

Cependant lorsque l'utilisateur connecté créé un fichier ou un dossier, ce dossier appartiendra au groupe courant désigné par la commande *newgrp*.

```
# pwd
/home/pierre/Desktop          # On est dans le répertoire de l'utilisateur pierre
# newgrp staff                # On change le groupe courant
# mkdir newDirectory          # On crée un nouveau dossier
# ls -la
total 12
drwxr-xr-x  3 pierre pierre 4096
drwxr-xr-x 14 pierre root   4096
drwxr-xr-x  2 pierre staff  4096 newDirectory
```

#Ce nouveau dossier appartient au groupe *staff*. Sans l'exécution de la commande il aurait appartenu au groupe *pierre*.

Cette appartenance du dossier au groupe *staff* sera inchangée jusqu'à la prochaine manipulation de droits.

## Synthèse

De la même façon qu'il est possible d'usurper l'identité d'un utilisateur, pour exécuter des actions en lieu et place de cet utilisateur, il est possible d'emprunter l'identité d'un groupe différent du groupe courant pour les mêmes raisons. Cela permet à un utilisateur donné ayant les droits de gérer un fichier, un dossier ou ensemble de dossiers pour le compte de tout un groupe dont il fait partie. Ainsi il n'est pas nécessaire de faire systématiquement appel à un administrateur ou au propriétaire "naturel" du groupe.

## Manipulation 7 : Gestion des mots de passe

L’administrateur peut accéder au mot de passe s’il est déclaré dans le fichier */etc/passwd*, il peut le changer en modifiant le fichier */etc/passwd*.

Sinon les mots de passe des utilisateurs qui sont stockés dans le fichier */etc/shadow* sont chiffrée. Un administrateur ne peut donc accéder qu'au hash du mot de passe. Pour l’obtenir il lui faut cracker ce mot de passe à partir du hash.

Il est cependant possible pour l'administrateur de réinitialiser le mot de passe d'un utilisateur avec la commande *passwd* en passant en mode **root**.

## Synthèse

Un mot de passe est strictement confidentiel et personnel. Il ne doit en aucun cas être communiqué à un tiers, pas même à un administrateur système. En effet, bien que cela soit une mauvaise pratique, on ne peut empêcher un utilisateur d'utiliser le même mot de passe sur plusieurs systèmes. Pour autant il faut que l'utilisateur *root* soit capable de gérer les cas de perte de mot de passe, d'où la possibilité de réinitialiser le mot de passe d'un utilisateur classique.

Même si l'utilisateur *root* a accès aux hashs des mots de passe et pas aux mots de passe eux-mêmes, cette information est malgré tout sensible : elle peut faciliter le craquage du mot de passe associé.

# Manipulation 8 : Mise en place de quota

Rôles des différentes commandes de base :

| **Commande** | **Rôle** |
|---|---|
| edquota | Permet l'attribution des quotas. La commande ouvre un éditeur qui permet de modifier directement les fichiers *aquotas.user* ou *aquota.group* |
| quota | Permet d'afficher l'utilisation et les limites du disque des utilisateurs. |
| repquota | Permet d'afficher un résumé des quotas et délais de grâce. |
| quoaton | Permet d'activer les quotas. |
| quotaoff | Permet de désactiver les quotas. |
| quotacheck | Cette commande permet d'initialiser la base des quotas afin d'éviter que les fichiers quotas deviennent incohérents. Ceci peut arriver après l'ajout d'un nouvel utilisateur ou groupe avec la commande *adquota*. |

## 1. Instaurez un quota

L'attribution de quotas dans un système de fichier permet de maitriser l'utilisation de l'espace disque en fixant des limites. Il existe deux types de limites :

La limite douce (*soft limit)* : quantité maximale d'espace qu'un utilisateur peut occuper. Si cette limite est atteinte, il reçoit un message d'avertissement.
Son utilisation est combinée avec les délais (grace period). Lorsqu'il continue à dépasser la *soft limit* après ce délai, il se retrouve dans le même cas que dans l'atteinte d'une limite dure.

La limite dure (*hard limit*) : limite absolue pour l'utilisation de l'espace. L'utilisateur ne peut pas dépasser cette limite. Passée cette limite, l'écriture sur ce système de fichiers lui est interdite.

⚠ Attention

Les limites sont exprimées en blocs et en inode. Exemple pour connaitre la taille des blocs dans un système de fichier :

```
# dumpe2fs /dev/sda1 | grep -i 'block size'
dumpe2fs 1.46.6-rc1 (12-Sep-2022)
Block size: 4096
```

Ici pour l'exercice la taille d'un bloc est de 4096 bits soit environ 216 octets.

- Installation du package quota :

```
# apt-get install quota
```

- Modification du fichier fstab

Ajout des options *ursquotas* pour les utilisateurs et *grpquota* pour les groupes :

```
# vi /etc/fstab
UUID=f02c792a-f423-4d70-b817-1bdc006babf2 / ext4
defaults,usrquota,grpquota,errors=remount-ro 0 1
```

- Remonter la partition :

```
# mount -vo remount /
mount: (hint) your fstab has been modified, but systemd
still uses the old version; use 'systemctl daemon-reload' to reload. mount:
/dev/sda1 mounted on /.
```

- Initialisation des quotas :

```
# quotacheck -vgum /
quotacheck: Scanning /dev/sda1 [/] done
quotacheck: Checked 38352 directories and 429749 files
```

-v : mode verbeux.
-u : vérifie seulement les quotas des utilisateurs présent dans */etc/mtab* ou dans le répertoire spécifié.
-g : vérifie seulement les quotas groupes présent dans */etc/mtab* ou dans le système de fichier spécifié.
-m : Ne pas remonter le système de fichier en lecture seule.

> ✎ Note
>
> Si la commande *quotacheck* n'est pas trouvé, le système propose de l'installer.

Vérification de la présence des fichiers avec les droits en lecture et écriture pour root uniquement :

```
# ls -l /
[...]
-rw------- 1 root root 9216 Apr 6 09:03 aquota.group
-rw------- 1 root root 9216 Apr 6 09:03 aquota.user
```

- Activation des quotas :

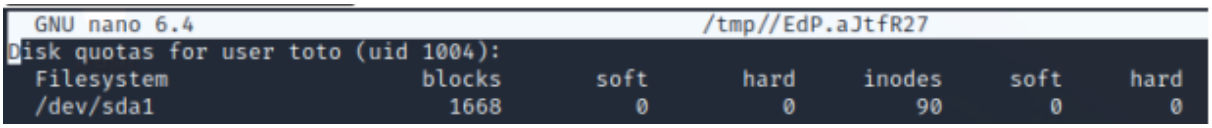
```
# quotaon -vgu /
/dev/sda1 [/]: group quotas turned on
/dev/sda1 [/]: user quotas turned on
```

- Attribution des quotas

Pour l'exercice, nous allons arbitrairement fixer les limites ci-dessous :

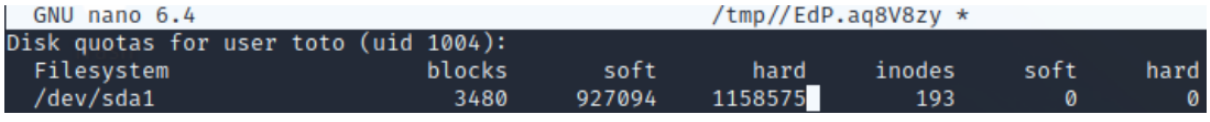
- limite douce : 200 Mo (environ 925926 blocs);
- limite dure : 250 Mo (environ 1157407 blocs).

```
# edquota -u toto
```




Ici, la taille pour l'utilisateur toto du dossier est de 1668 blocs (soit environ 0,36Mo)

Pour respecter les quotas précédemment établis, nous les ajoutons, ce qui donne :



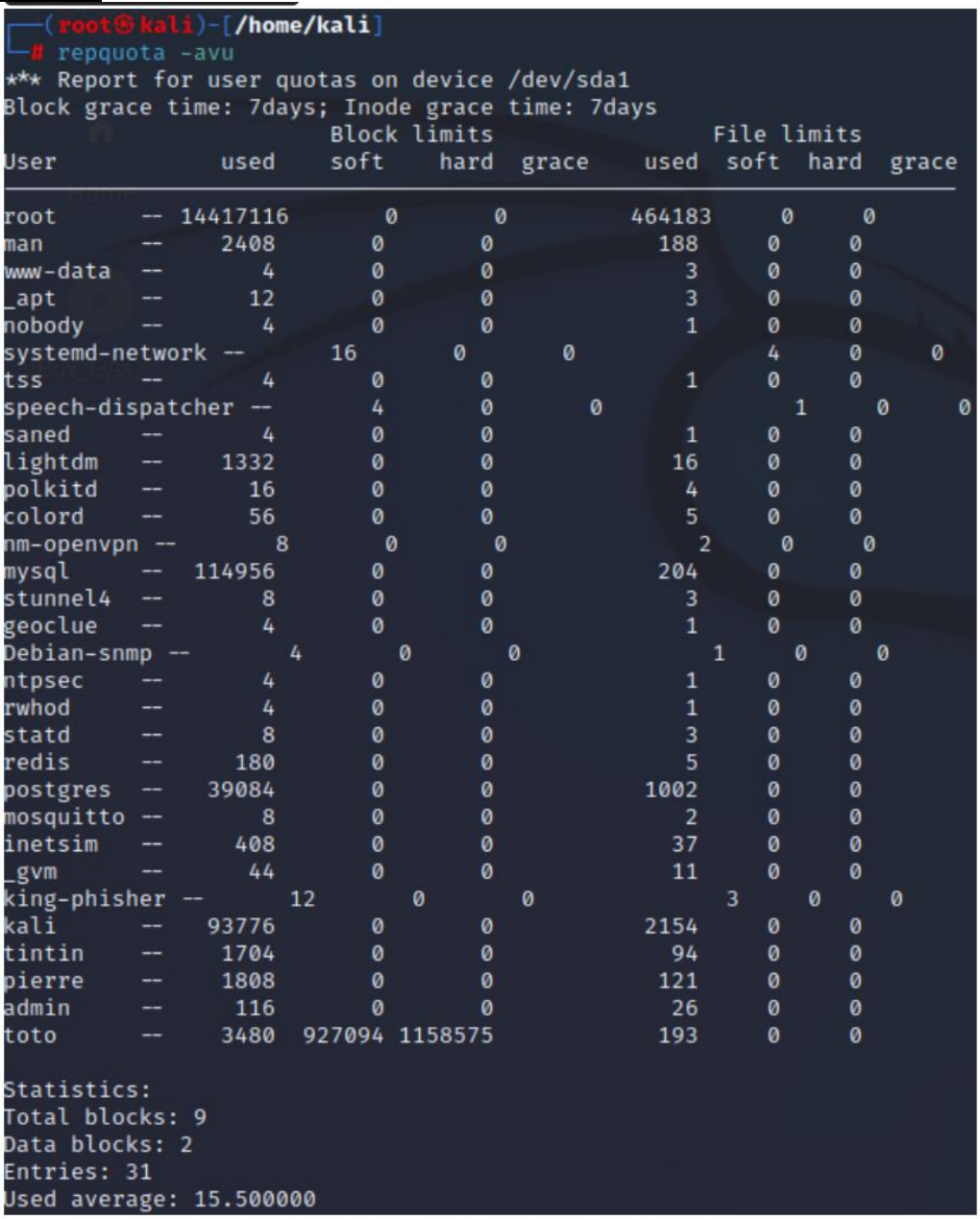

Fixation des quotas :

```
# edquota -t
```

Par défaut il est de 7 jours.

- Visualisation des quotas :

```
# repquota -avu
```



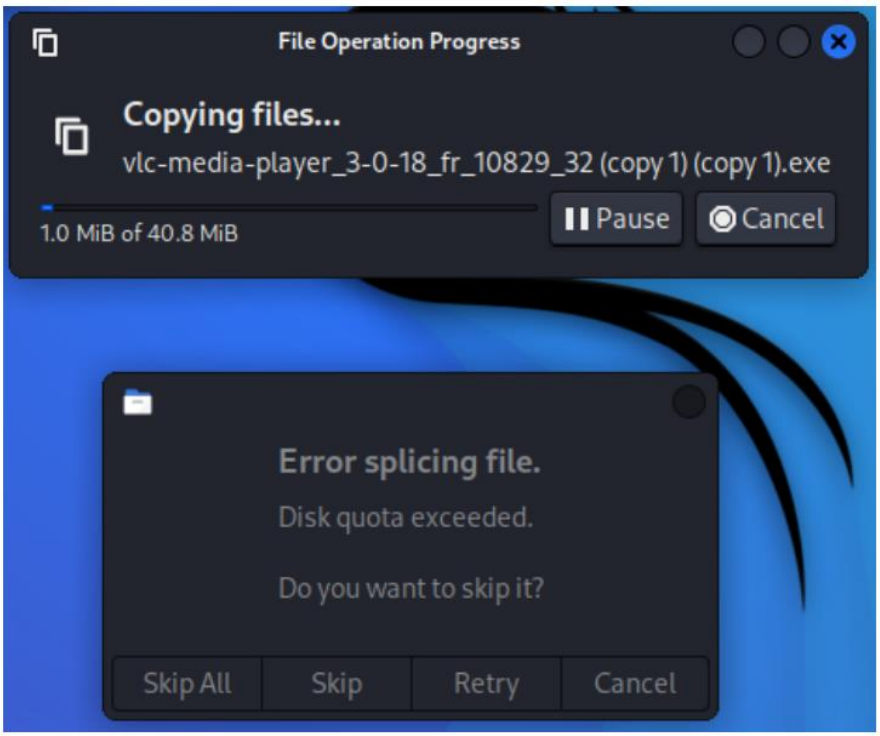
```
┌──(root㉿kali)-[/home/kali]
└─# repquota -avu
*** Report for user quotas on device /dev/sda1
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root       -- 14417116       0       0         464183     0     0
man        --    2408       0       0            188     0     0
www-data   --       4       0       0              3     0     0
_apt       --      12       0       0              3     0     0
nobody     --       4       0       0              1     0     0
systemd-network --      16       0       0              4     0     0
tss        --       4       0       0              1     0     0
speech-dispatcher --       4       0       0              1     0     0
saned      --       4       0       0              1     0     0
lightdm    --    1332       0       0             16     0     0
polkitd    --      16       0       0              4     0     0
colord     --      56       0       0              5     0     0
nm-openvpn --       8       0       0              2     0     0
mysql      --  114956       0       0            204     0     0
stunnel4   --       8       0       0              3     0     0
geoclue    --       4       0       0              1     0     0
Debian-snmp --       4       0       0              1     0     0
ntpsec     --       4       0       0              1     0     0
rwhod      --       4       0       0              1     0     0
statd      --       8       0       0              3     0     0
redis      --     180       0       0              5     0     0
postgres   --   39084       0       0           1002     0     0
mosquitto  --       8       0       0              2     0     0
inetsim    --     408       0       0             37     0     0
_gvm       --      44       0       0             11     0     0
king-phisher --      12       0       0              3     0     0
kali       --   93776       0       0           2154     0     0
tintin     --    1704       0       0             94     0     0
pierre     --    1808       0       0            121     0     0
admin      --     116       0       0             26     0     0
toto       --    3480  927094 1158575            193     0     0

Statistics:
Total blocks: 9
Data blocks: 2
Entries: 31
Used average: 15.500000
```

Nous observons que l'utilisateur *toto* ne dépasse pas sont quotas.

- Remplissage du compte *toto* pour dépasser le quota :

Au bout d'un moment nous avons le message suivant « Disk quota exceeded » :




Il est impossible à l'utilisateur "tintin" d'ajouter le moindre fichier.

> ⓘ Info
>
> Nous n'avons pas eu l'apparition d'un message lors du dépassement de la limite douce.

Vérification avec la commande *repquota -avu* :

```
toto +- 1158580 927094 1158575 6days 536 0 0
```

Nous pouvons observer le dépassement de la limite dure des blocs +-. Le - correspondant à la limite d'inodes.

> 🖉 Note
>
> L'utilisation de la commande `quotatool` permet de s'affranchir du calcul du nombre et taille des blocs. Il est possible de préciser l'unité : k(kilo), m(mega), g(giga)
>
> ```
> # quotatool -u <user> -bq <SoftLimit> -l <HardLimit> /home
> ```

### 2. Désactivation des quotas

- Remplissage du compte de l'utilisateur

Nous mettons suffisamment de fichier pour dépasser le quota mis précédemment.

Utilisation de la commande *quotacheck* permettant de synchroniser les fichiers pour que la prise en compte se fasse correctement :

```
# quotacheck -auvgm
quotacheck: Scanning /dev/sda1 [/] done
quotacheck: Checked 38485 directories and 431576 files
```

### 3. Réactivation des quotas

```
# quotaon -vgu /
```

Il est impossible de se connecter au compte "toto" avec l'interface graphique.

Possibilité de se connecter en utilisant la commande *su*.

Pour supprimer les fichiers il est nécessaire d'utiliser la commande *rm -f*, sinon nous avons le message suivant :

```
rm: cannot remove write-protected regular file '/home/toto/test.exe'?
```

Cela permet de pouvoir redescendre en dessous des quotas et de se reconnecter sur le compte toto.

L'ensemble de l'exercice a été effectuer sur la racine / directement car nous n'avons pas réussi à créer un système de fichier et appliquer le quota uniquement sur le dossier /home/toto.

### 4. Quota de groupe

Un quota sur un groupe à le même rôle que pour un utilisateur. Il permet lors de l'administration d'attribuer directement à tous les utilisateurs d'un groupe l'attribution d'une quantité d'espace disque.

## Synthèse

L'utilisation des quotas dans une entreprise est primordiale. Car l'espace de stockage n'est pas infini, cela peut certes évoluer en rajoutant de la mémoire de stockage mais ça un coût financier non négligeable. Cette mise en place permet à chaque utilisateur de gérer son espace de stockage qui lui est attribué, de façon individuelle ou pour les groupes associés. Sans cela il y aurait des inégalités de l'utilisation de l'espace, mais surtout cela ralentirait et impacterai le réseau de l'entreprise, voir empêcher un utilisateur de se connecter.

Manipulation 9 : ACL

Création du compte tintin avec la commande *adduser*. Puis création du fichier avec la commande *touch*.

```
# touch article.txt                          # Création du fichier
# setfacl -m u:tintin:rw- article.txt        # Attribution des droits en écriture et
                                             lecture pour tintin sur le fichier
# getfacl article.txt                        # Récapitulatif des permissions sur ce fichier
# file: article.txt
# owner: root
# group: root
user::rw-
user:tintin:rw-
group::r--
mask::rw-
other::r--
```

L'utilisateur tintin à des droits en lecture et écriture sur ce fichier indiqué avec le ***rw-***.

```
# setfacl -m m:r--,o:r-- article.txt          # Application d'un mask de lecture seule
# getfacl article.txt
# file: article.txt
# owner: root
# group: root
user::rw-
user:tintin:rw- #effective:r--
group::r--
mask::r--
other::r--
```

À la suite de la commande, les autres utilisateurs et groupes ont seulement le droit de lecture sur le fichier indiqué avec le ***r--***.

```
# mkdir PetitReporter                        # Création du répertoire "PetitReporter"
```

Application des permissions

```
# setfacl -Rm d:u:tintin:rw-,d:g::---,d:m:---,d:o:--- PetitReporter/
# getfacl PetitReporter/
# file: PetitReporter/
# owner: root
# group: root
user::rwx
user:tintin:rw-
group::r-x
mask::rwx
other::r-x
default:user::rwx
default:user:tintin:rw- #effective:---
default:group::---
default:mask::---
default:other::---
```

Seul tintin a des droits sur ce dossier.

```
# touch PetitReporter/autreArticle.txt    # Création d'un fichier dans le dossier
# getfacl PetitReporter/autreArticle.txt
# file: PetitReporter/article.txt
# owner: root
# group: root
user::rw-
user:tintin:rw- #effective:---
group::---
mask::---
other::---
```

Ce fichier a bien hérité des permissions du dossier.

```
# setfacl -b article.txt
# getfacl article.txt
# file: article.txt
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

L'option -b permet de supprimer l'ACL de tout le fichier.

```
# setfacl -m u:tintin:rwx,g::rwx article.txt          # On reset des permissions
# getfacl article.txt
# file: article.txt
# owner: root
# group: root
user::rw-
user:tintin:rwx
group::rwx
mask::rwx
other::r--
```

```
# setfacl -x u:tintin article.txt
# getfacl article.txt
# file: article.txt
# owner: root
# group: root
user::rw-
group::rwx
mask::rwx
other::r--
```

On voit bien que l'option -x permet de supprimer seulement les droits de l'utilisateur sélectionné.

## Synthèse

Les ACL permettent une gestion des droits plus riche qu'un simple *chmod*. Elle permet l'application d'un mask pour configurer des droits par défaut et permet d'appliquer un héritage sur un dossier pour ne pas avoir à paramétrer individuellement chaque fichier ou dossier.

# Synthèse globale

Un système cohérent et bien pensé de droit utilisateur est une composante essentielle d'une configuration système assurant la cohérence et la sécurité des données des différents intervenants travaillant sur ce système. Cette configuration consiste en la création d'utilisateurs et de groupes cohérents. Elle est faite dans un certain nombre de fichiers bien précis, chacun ayant un rôle défini et comportant des informations plus ou moins sensibles.

La création d'utilisateur ou de groupe peut se faire selon différentes méthodes, plus ou moins précises, plus ou moins de haut ou de bas niveau, plus ou moins sécurisées.

L'administration de tels comptes utilisateur permet de les bloquer ou de les débloquer pour, par exemple, des raisons de sécurité et de sensibilisation auprès des utilisateurs travaillant sur le système. On peut aussi supprimer un compte et conserver ou non les données, donner un accès temporaire, forcer la modification de mot de passe, attribuer des droits aux utilisateurs ou aux groupes...

Il existe plusieurs types de compte. Les comptes utilisateurs bien sûr, mais aussi des comptes systèmes tels que ***root*** ou ***bin***.

Le compte ***root*** dispose de tous les droits de façon illimité. C'est aussi un compte qui rend le système totalement vulnérable s'il est hacké. Plus que jamais pour ce compte-là, la sécurité doit être maximale (mot de passe fort, qui ne soit pas noté quelque part, toujours bien se déconnecter de sa session avant de quitter l'ordinateur...)

Les comptes systèmes autre que *root* disposent de permissions spéciales pour exécuter différentes tâches nécessitant des droits élevés mais pas forcément de privilèges *root*. Autant que possible il faut éviter d'exécuter des opérations en tant qu'administrateur root.

Viennent ensuite les comptes utilisateurs, garantissant à chacun un accès sécurisé et privé au système, un espace de travail réservé et une configuration système personnalisée. Les utilisateurs classiques peuvent se voir attribuer des privilèges d'administration.

Un compte linux est caractérisé par un certain nombre de données, telles que son UID, son GID, son mot de passe, son répertoire de connexion, le shell utilisé, etc... Ces données sont présentes dans le fichier */etc/passwd*. D'autres informations le concernant sont stockées dans */etc/shadow* où le mot de passe est crypté (le hash de ce dernier), la date de sa dernière modification, la date où il devra être changé, etc... Ces informations permettent le blocage d'un compte ou son déblocage. Ces informations peuvent aussi être mises à disposition grâce à un serveur NIS qui est un service d'annuaire définissant les différentes connexions.

L'utilisation d'un mot de passe robuste est essentielle à la sécurité d'un compte linux. Un mot de passe sécurisé ne doit pas être déductible d'informations concernant son propriétaire, doit contenir au moins 8 caractères avec des majuscules, des chiffres et des caractères spéciaux. Il existe des programmes permettant de régulièrement tester l'efficacité des mots de passe (par exemple en tentant de les casser avec une attaque par force brute). De tels programmes sont paramétrables pour par-exemple bloquer les comptes présentant des vulnérabilités et obliger l'utilisateur à modifier son mot de passe pour un nouveau plus résistant. Bien que cela puisse être très utile, la meilleure des solutions reste la sensibilisation, de nombreuses failles de sécurité liés aux façons de faire des utilisateurs sont autant de porte d'entrée potentielles pour un attaquant.

De la même façon que pour un compte utilisateur, un groupe est caractérisé par son GID ou la liste de ses membres. Différentes méthodes existent également pour créer un nouveau groupe ou le modifier. Un groupe permet de regrouper plusieurs utilisateurs qui vont avoir des caractéristiques et des droits en commun. Cette organisation permet de mutualiser le travail tout en garantissant la sécurité et la confidentialité des espaces de travail de groupes sur le système malgré la connexion d'utilisateurs étrangers à ces groupes.

La gestion par utilisateurs ou par groupes permet aussi de mieux maîtriser l'accès aux ressources (mémoire, nombre de fichiers). Il est possible de mettre des quotas sur ces entités mais aussi sur un disque, des i-nodes. Les quotas servent de garde-fous. Il est possible de vérifier si les quotas sont respectés, de définir une limite produisant un warning pour que l'utilisateur ne se retrouve pas coincé sans s'y attendre. Ainsi cela garanti que tout le monde bénéficie des ressources dont il a besoin pour travailler dans les meilleures conditions possibles.

Enfin, une composante essentielle de l'administration d'utilisateurs et des groupes sont l'attribution de permission. Les permissions concernent les droits de lecture, d'écriture ou d'exécution d'un fichier, ou l'accès à un dossier. Là aussi des commandes spéciales permettent de configurer ces droits. Parmi ces méthodes figure l'Access Control List (ou ACL) qui est une surcouche permettant une gestion plus fine des droits. L'ACL permet entre-autres l'héritage des droits dans un dossier donné pour en simplifier l'administration.