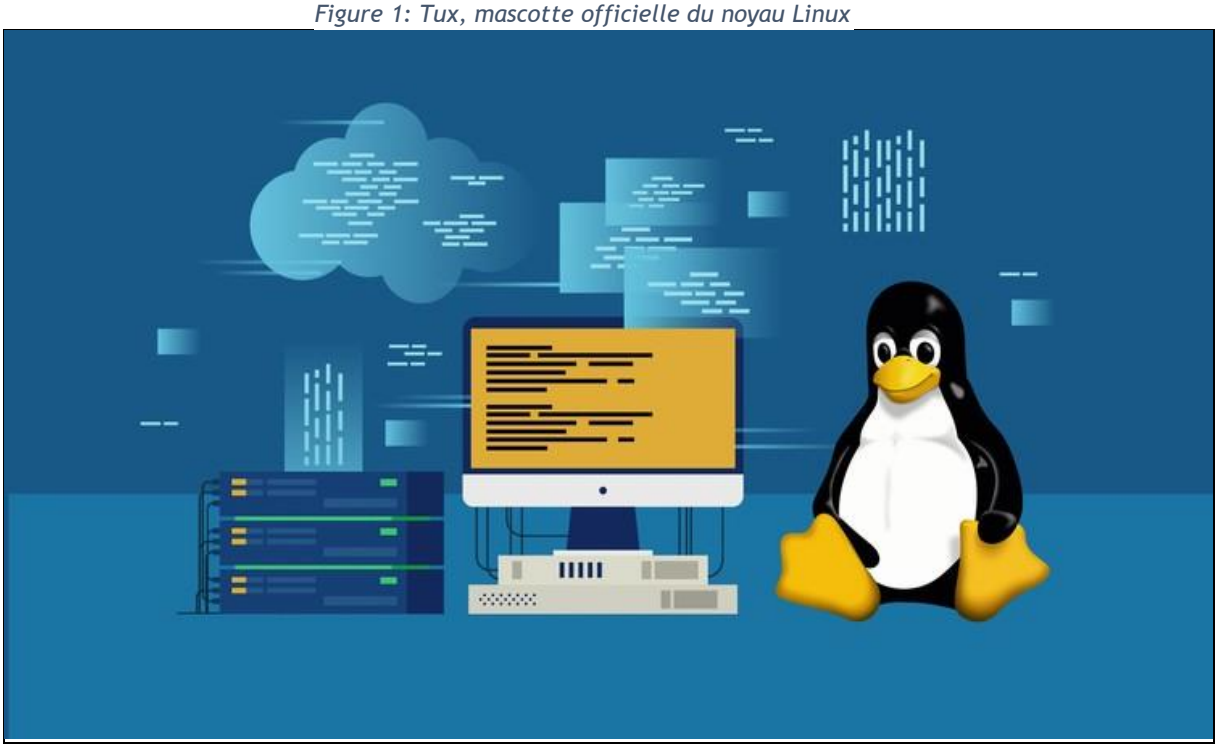
Hes·so VALAIS WALLIS
Haute Ecole de Gestion
Hochschule für Wirtschaft

Filière Informatique de gestion
Studiengang Wirtschaftsinformatik

Projet intermédiaire Raspberry Pi

Configurations de la sécurité de base des comptes et des accès aux fichiers et répertoires

Figure 1: Tux, mascotte officielle du noyau Linux



Source : (formation-installation-et-administration-dun-serveur-linux, 2023

Etudiant : Dasek Joiakim

Professeurs : Barmaz Xavier, Russo David

Date de rendu : 05.01.2023

Résumé

Voici le plan du document : nous allons d'abord introduire le sujet de la sécurité et l'efficacité de « Linux » dans ce domaine. Ensuite, nous allons comprendre l'importance de configurer et de sécuriser les comptes et accès aux fichiers et répertoires pour garder une intégrité des données, puis nous allons voir comment créer et gérer des comptes d'utilisateurs ainsi que les groupes et leurs efficacités.

Ensuite, nous aborderons la gestion des droits d'accès aux fichiers et répertoires, en expliquant comment utiliser les commandes « chmod », « chown » ou encore « chgrp ». Enfin, nous allons voir de loin, quelques autres moyens pour sécuriser les fichiers ou dossiers, tout au long du document, nous allons mettre en pratique la théorie avec des scénarios que nous pouvons rencontrer dans un parcours professionnel.

Nous sommes maintenant prêts à débuter, passons à la première partie de notre apprentissage : l'introduction à la sécurité de base des comptes et accès aux ressources.

Mots clés : useradd, usermod, userdel, chmod, chown, chgrp, grpadd, su, GPG

Table des matières

Liste des figures

Liste des abréviations

- ERP — Entreprise Resource Planning, est un type de logiciel de gestion qui permet à une entreprise de gérer l'ensemble de ses opérations.
- GIT — C'est un logiciel de gestion de version de code informatique.
- Backend / Frontend — Ce sont deux parties d'une application, respectivement coté serveur et client.
- GUI — Graphical User Interface, Interface graphique pour utilisateur.
- useradd — Commande pour ajouter un utilisateur (Linux).
- passwd — Commande pour créer ou modifier un mot de passe (Linux).
- usermod — Commande pour modifier des informations lié à un utilisateur (Linux).
- userdel — Commande pour supprimer un utilisateur (Linux).
- deluser — Enlever un utilisateur dans un groupe (Linux).
- grpadd — Commande pour créer un groupe (Linux).
- grpdel — Commande pour supprimer un groupe (Linux).
- man — Manual, commande pour consulter de la documentation sur d'autres commandes et outils (Linux).
- UID — User ID, identifiant de l'utilisateur.
- GID — Groupe ID, identifiant de groupe.
- cut — Commande pour récupérer du contenu d'un fichier avec un délimiteur et un numéro de colonne (Linux).

- su — Switch User, commande pour changer d'utilisateur (Linux).
- grep — Commande qui permet de filtrer du contenu selon un paramètre donné (Linux).
- Passphrase — A la différence d'un mot de passe, il s'agit d'une phrase de passe.
- chmod — Change mode, commande pour changer les droits d'une ressource (Linux).
- chown — Change owner, commande pour changer l'utilisateur ou groupe propriétaire d'une ressource (Linux).
- chgrp — Change groupe, commande pour change de groupe propriétaire d'une ressource (Linux).
- GPG — GNU Privacy Guard, logiciel de chiffrement.
- RBAC — C'est un modèle de contrôle d'accès qui autorise l'accès à une ressource informatique en fonction des rôles.
- MAC — C'est un modèle de contrôle d'accès à une ressource informatique en fonction de la sécurité de l'objet ainsi que de l'utilisateur.

1. Introduction à la sécurité de base des comptes et accès aux ressources

Ce document présente l'aspect de la sécurité sur la configuration des comptes, des accès aux fichiers et répertoires sous Linux. Linux est un système d'exploitation open source, c'est-à-dire que son code est accessible et modifiable par tous. Cette caractéristique contribue à la robustesse de ce système et permet à la communauté de participer à son enrichissement et à son amélioration en constante progression.

Tout système informatique n'est sécurisé à cent pour cent ! La gestion des ressources et la protection des données sont essentielles dans un système informatique. Sous Linux, la personnalisation élevée offre la possibilité de configurer et de sécuriser ces éléments selon nos besoins. Nous allons donc voir comment configurer et sécuriser ces éléments.

Nous verrons comment créer et gérer des comptes d'utilisateurs, et comment gérer les accès aux fichiers et répertoires. En dernier point nous aurons un exemple pratique d'une situation réelle dont le besoin est d'appliquer ces aspects de sécurité.

2. Pourquoi est-il important de configurer et de sécuriser les comptes et accès aux fichiers et répertoires ?

Il existe plusieurs raisons de protéger nos systèmes d'informations contre les utilisateurs non autorisés, notamment en mettant en place une configuration de comptes d'utilisateurs et en gérant les permissions d'accès aux ressources de manière à les empêcher de lire, d'écrire ou de supprimer des données sensibles.

Par exemple, si nous avons une équipe de développeurs pour une application interne de type « ERP » qui utilise « GIT » comme logiciel de gestion de versions et que chaque membre de cette équipe a des niveaux de maîtrise différents dans le métier, comme un apprenti, ainsi que des rôles différents, comme un développeur « backend » ou « frontend ».

Il est alors judicieux et même crucial que chaque personne appartienne à un groupe spécifique. Par exemple, les développeurs seniors pourraient être liés à un groupe senior avec un accès et un contrôle plus étendu des ressources, alors que le groupe apprentis ne verrait que certaines actions possibles sur le système.

Des actions de suppression définitive de code source ou de poussée de code source, en production, par erreur peuvent être dangereuses et entraîner des répercussions importantes pour la vie économique de l'entreprise, la notoriété, l'image de marque etc…

En plus de la restriction des comptes, une autre raison de configurer et de sécuriser les accès aux fichiers et répertoires est de garantir la confidentialité et l'intégrité des données, comme une base de données stockée sur le système. Pour éviter un accès de n'importe qui, nous pourrions définir des mots de passe robustes ou utiliser l'authentification à plusieurs facteurs.

3. Comment créer et gérer les comptes d'utilisateurs ?

Nous allons maintenant voir comment créer et gérer des utilisateurs sous Linux en ligne de commande. La raison de cela est que l'utilisation de la ligne de commande est plus portable et adaptée aux différentes distributions Linux, contrairement aux interfaces graphiques qui peuvent varier fréquemment et être moins personnalisables en termes d'exploitation des commandes que nous verrons.

Nous allons maintenant passer à la pratique. Pour accéder à notre compte par défaut sur le Raspberry Pi, nous devons nous connecter avec le nom d'utilisateur "pi". Une fois connectés, si nous n'avons pas accès directement à l'interface graphique, nous pouvons rester sur la console. Si nous sommes redirigés vers l'interface graphique après la connexion, nous devrons ouvrir le terminal.

Il est important de noter que la console est l'interpréteur natif des commandes que nous entrons, tandis que le terminal est une émulation de la console accessible via l'interface graphique (GUI).

Nous allons tout d’abord apprendre les commande « useradd », « passwd », « usermod » et enfin « userdel » qui signifie respectivement :

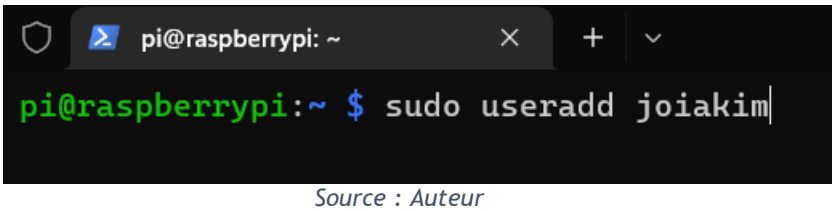
- Ajouter un utilisateur
- Définir ou modifier un mot de passe
- Modifier les informations d’un compte existant
- Supprimer un compte

Nous pouvons précéder à l'une de ces commande le terme « man » pour avoir plus d'amples informations sur ce qu'est cet outil et comment l'utiliser, avec des options spécifiques à apprendre au besoin.

1. La commande « useradd »

Voici comment ajouter un nouvel utilisateur, ne pas oublier de précéder du mot-clé « sudo » parce qu'il faut les privilèges administrateur pour exécuter cette commande : « sudo useradd nomutilisateur » on peut ajouter l'option « -d /home/userX » qui spécifie la création d'un répertoire « home », si ce n'est pas spécifier alors le système enverra l'utilisateur créé dans le répertoire de « pi » mais n'aura aucun droit dessus :

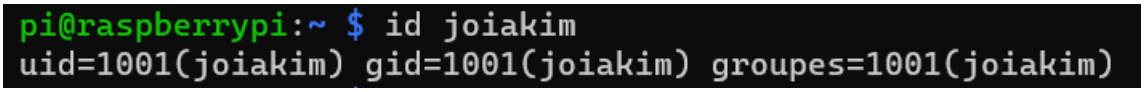
Figure 2 - Commande, ajouter un utilisateur



Source : Auteur

Si tout s'est bien déroulé alors le terminal ne répondra rien, ainsi pour vérifier que l'utilisateur a été ajouté nous pouvons exécuter la commande suivante « id joiakim » et il nous répondra de la manière suivante :

Figure 3 - Commande id, pour obtenir les détails utilisateur



Source : Auteur

Le « UID » est l'identifiant de l'utilisateur, le « GID » est l'identifiant de groupe nous allons aborder ce que sont les groupes dans la suite du document.

Nous pouvons aussi lister tous les utilisateurs ainsi : « sudo cut -d: -f1 /etc/passwd » :

Figure 3 - Commande pour ressortir les utilisateurs existants

Source : Auteur

La commande « cut » permet d'extraire des parties de chaque ligne d'un fichier pour filtrer le contenu et rendre lisible ce qui nous intéresse, ses paramètres qui suivent sont là pour éliminer tout ce qui suit après le nom d'utilisateur. Nous pouvons utiliser cette commande sans les paramètres ce qui reviendrait à faire « sudo nano /etc/passwd » avec tous les détails de chaque compte.

En règle générale la commande « cut » est utilisé avec un délimiteur et un numéro de colonne, respectivement « -d » et « -f », par exemple ci-dessus « cut -d: -f1 /etc/passwd ». Cela nous permet de couper chaque ligne avec comme délimiteur le « : » et récupérer uniquement la première colonne. Dans le fichier « etc/passwd » la ligne où se trouve le

compte « joiakim » il est écrit « joiakim:1234:groups:xyz… » c'est pourquoi « joiakim » est uniquement récupéré.

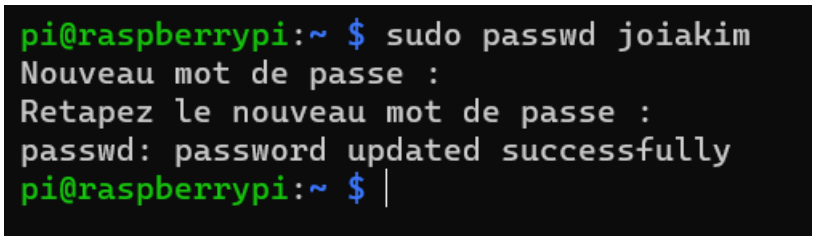
Le compte n'est pas accessible pour l'instant parce qu'aucun mot de passe ne lui a été définie, pour preuve nous pouvons changer d'utilisateur « switch user » avec la commande « su joiakim » et le terminal nous demande un mot de passe pourtant aucun ne lui a été attribuer. Cela veut dire qu'il est nécessaire de définir un mot de passe à celui-ci.

2. La commande « passwd »

C'est une commande qui permet de définir le mot de passe de l'utilisateur, elle nécessite des privilèges et permettra d'écrire le mot de passe chiffré dans « /etc/shadow ».

Nous utilisons donc la commande « passwd » suivis du nom de l'utilisateur :

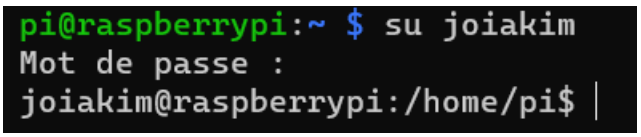
Figure 4 - Commande pour définir un mot de passe



Source : Auteur

Nous pouvons dès à présent changer d'utilisateur pour voir si cela fonctionne « su joiakim » :

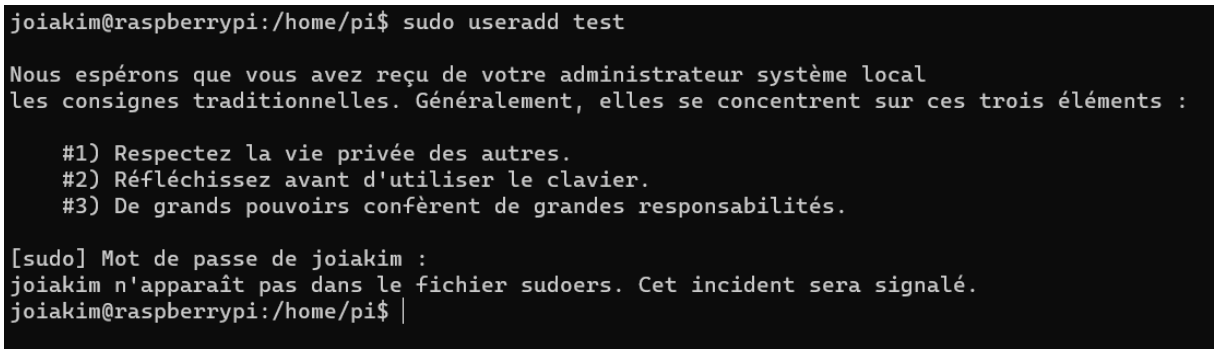
Figure 5 - Commande pour changer d'utilisateur



Source : Auteur

On constate que nous sommes bien connectés au compte, voyons si nous pouvons effectuer la même démarche pour créer un nouveau compte depuis « joiakim » :

Figure 6 - Commande pour ajouter un utilisateur



Source : Auteur

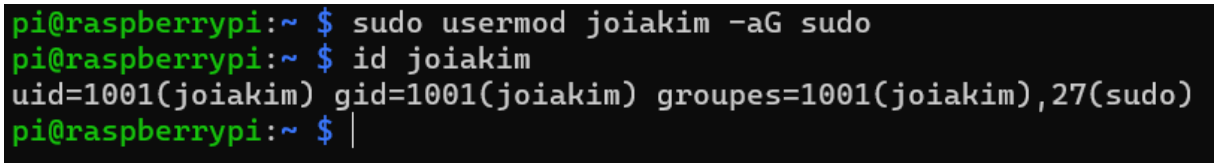
Nous exécutons la commande et directement un avertissement nous parvient, nous demandant d'entrer le mot de passe de l'utilisateur courant. Il nous répond « joiakim n'apparaît pas dans le fichier sudoers. Cet incident sera signalé » voilà une couche de sécurité mise en place par défaut par le système quand nous créons un utilisateur sans spécifier de configuration. Nous aborderons ce sujet par la suite. Exécutons la commande « exit » pour sortir de la session actuelle et rejoindre la session « pi » active.

3. La commande « usermod » et « userdel »

Nous utilisons à présent « usermod » pour modifier les données de l'utilisateur et nous allons lui accorder l'utilisation de la commande « sudo ». Attention ! l'utilisation de cette commande offre une liberté à l'utilisateur qui la possède, à accorder avec grande vigilance et il n'est pas courant de permettre à n'importe qui de l'utiliser.

A titre d'exemple nous allons voir comment le faire « sudo usermod joiakim -aG ». Le paramètre « a » pour « append » donc attacher l'utilisateur à un group « G ». L'utilisation du paramètre « a » est indissociable du « G ». Le paramètre « G » requiert de spécifier quel groupe, en l'occurrence le groupe « sudo » qui ont droit à son utilisation :

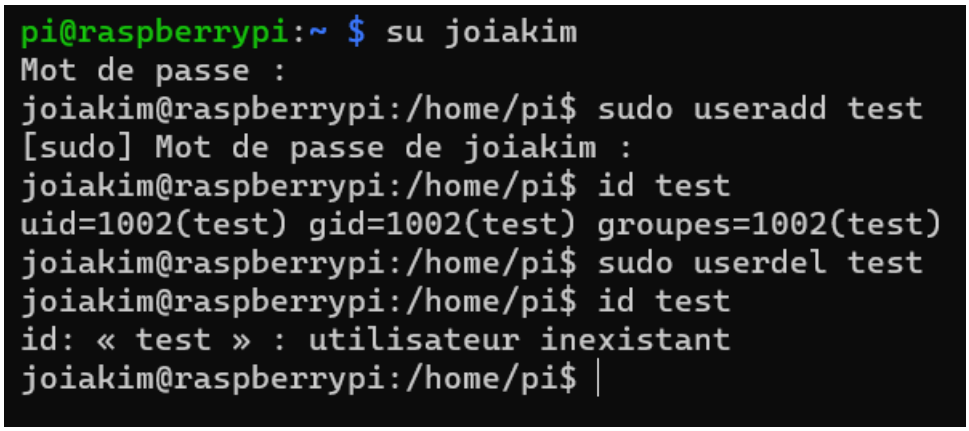
Figure 7 - Commande pour ajouter un groupe secondaire



Source : Auteur

Nous remarquons donc avec la commande « id » un nouveau groupe avec l'identifiant « 27 ». Afin de lire une fois littéralement ce qui est écrit cela donnerait : « Joiakim à identifiant « 1001 » appartient au groupe principal « 1001 » et au groupe normal « 27 ». Nous avons donc la confirmation que « joiakim » peut exécuter la commande « sudo ». Nous allons tout de même nous connecter au compte « joiakim » et effectuer la commande pour créer un utilisateur afin d'utiliser la commande de suppression de compte par la suite, voici la commande à exécuter pour supprimer un utilisateur « userdel » comme suit :

Figure 8 - Commande pour changer d'utilisateur



Source : Auteur

4. Les groupes principaux et secondaires

Nous allons introduire ce qu'est un groupe, comme son nom l'indique et en l'occurrence sur « Linux », un groupe est un ensemble d'utilisateurs partageant les mêmes autorisations d'accès aux fichiers et des ressources du système. Comme vu précédemment, pour chaque utilisateur, un seul groupe principal lui est associé et zéro ou plusieurs autres groupes secondaires peuvent lui être assigné.

Les groupes sont très utiles pour gérer les autorisations d'accès aux fichiers et aux ressources sur le système. Pour reprendre l'exemple de l'introduction, si nous avons une équipe de développeurs qui travaille sur le projet « ERP », il est important de bien gérer les autorisations d'accès aux fichiers et aux ressources du système. Pour ce faire, nous allons utiliser les groupes de manière à faciliter l'accès aux ressources pour l'ensemble de notre équipe de développeurs.

Nous allons donc créer un groupe "dev-senior" et y ajouter tous les comptes d'utilisateurs de l'équipe qui travaillent sur ce projet. Cela nous permettra de donner au groupe "dev-senior" les autorisations d'accès aux fichiers et aux répertoires nécessaires pour le projet, de manière que tous les membres de l'équipe puissent y accéder sans avoir à configurer individuellement les autorisations pour chaque compte d'utilisateur.

En utilisant les groupes de cette manière, nous garantissons une meilleure gestion des autorisations d'accès aux ressources du système, tout en simplifiant l'accès pour l'ensemble de notre équipe de développeurs. Les groupes permettent aussi de restreindre l'accès aux ressources.

5. Partie pratique

Afin de rendre cela plus compréhensible, notre équipe de développeurs contient des développeurs séniors et des apprentis. Nous avons deux groupe distincts « dev-senior » et « apprenti » et nous connaissons le groupe « sudo » qui permet à ses utilisateurs d'exécuter la commande « sudo » donc à ne pas assigner à n'importe qui.

Le but ici est de comprendre comment les groupes font hériter d'une liberté ou d'une restriction.

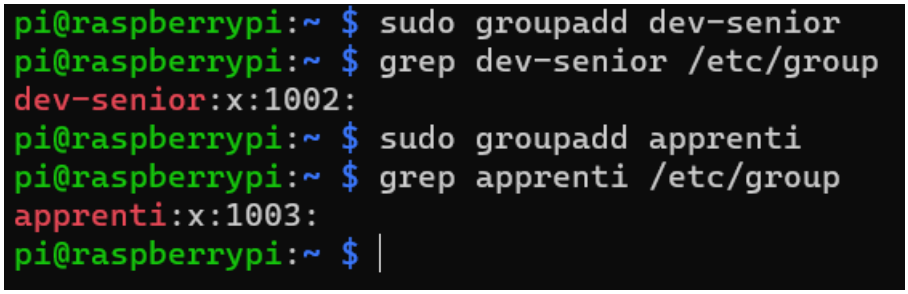
La manipulation sur le terminal suivra ces étapes :

1. Créer un groupe « dev-senior » et « apprenti ».
2. Créer les deux utilisateurs « maxime » et « leo ».
3. Assigner des mots de passes pour chacun.
4. Attacher le groupe secondaire « dev-senior » et le deuxième groupe secondaire « sudo » pour que l'utilisateur « maxime » ait toujours comme groupe principal son groupe éponyme. Il aura donc accès à ses ressources propres.
5. Assigner ces utilisateurs à ces groupes et effectuer une action sudo sous chaque compte pour observer l'efficacité des groupes.

En réalité, avant même de toucher au terminal une analyse approfondie de l'utilisateur doit être faite. Que fait-il sur le système d'information et ainsi commencer à restreindre avant même de lui donner plus de liberté, et si l'on n'est pas sûr, on pourrait ajouter aux libertés accordées, d'imposer à ce que l'utilisateur accompagne sa commande d'un mot de passe pour s'authentifier. Beaucoup d'options s'offrent à nous !

Tout d'abord nous créons les deux groupes avec la commande « sudo groupadd dev-senior » et « apprenti » en vérifiant avec la commande « grep » qui permet de rechercher les occurrences d'un terme ou d'une expression régulière dans un fichier :

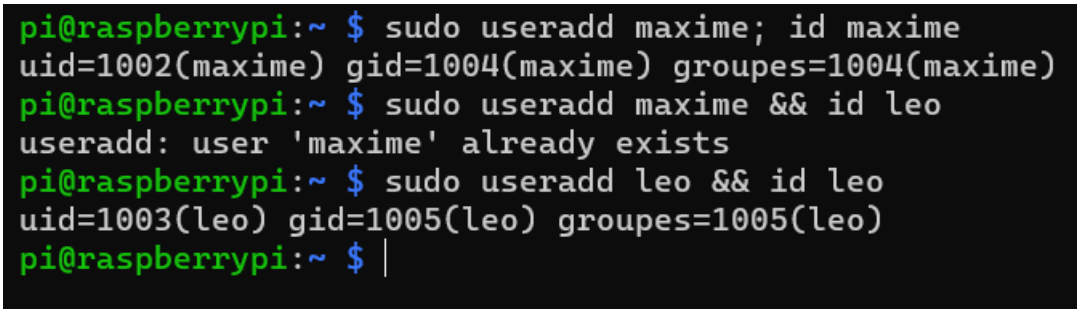
Figure 9 - Commande pour créer un groupe



Source : Auteur

Créons les deux utilisateurs et vérifions leurs existences :

Figure 10 - Commande pour créer un utilisateur



Source : Auteur

Dans cette figure nous connaissons toutes les commandes, je vais tout de même expliquer les opérateurs logiques « && » ainsi que « | » et le point-virgule entre deux expressions.

Premièrement le point-virgule, comme en programmation, permet d'indiquer à l'interprétateur ou compilateur en l'occurrence ici interprétateur de commande de savoir que c'est la fin d'une commande. On peut donc enchaîner une nouvelle commande sur la même ligne.

Nous avons l'opérateur logique « && » qui se veut être « AND » seulement si l'expression de gauche ne retourne aucune erreur donc est « vraie », on peut exécuter la commande de droite !

Enfin l'opérateur logique « || », uniquement si l'expression de gauche est fausse alors on exécute celle de droite. Nous pouvons observer ces comportements dans la figure ci-dessus.

Nous devons assigner un mot de passe à chacun des utilisateurs :

Figure 11 - Commande pour créer un mot de passe

```
pi@raspberrypi:~ $ sudo passwd maxime && sudo passwd leo
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
pi@raspberrypi:~ $
```

Source : Auteur

Continuons notre exercice, nous devons rattacher les deux groupes secondaires, « dev-senior » et « sudo » à « maxime » et le groupe secondaire à « leo » avec la commande « sudo usermod -aG sudo maxime && sudo usermod -aG dev-senior maxime && id maxime; sudo usermod -aG apprenti leo && id leo » et vérifiant que cela est bien appliqué :

Figure 12 - Commande pour ajouter un groupe secondaire

```
pi@raspberrypi:~ $ sudo usermod -aG sudo maxime && sudo usermod -aG dev-senior maxime
pi@raspberrypi:~ $ id maxime
uid=1002(maxime) gid=1004(maxime) groupes=1004(maxime),27(sudo),1002(dev-senior)
pi@raspberrypi:~ $ sudo usermod -aG apprenti leo && id leo
uid=1003(leo) gid=1005(leo) groupes=1005(leo),1003(apprenti)
pi@raspberrypi:~ $
```

Source : Auteur

Nous devons ainsi tester l'utilisation de la commande sudo pour l'utilisateur « maxime » et « leo » :

Figure 13 - Commande pour changer d'utilisateur et tester l'effet des groupes

```
pi@raspberrypi:~ $ su leo
Mot de passe :
leo@raspberrypi:/home/pi$ sudo useradd test
[sudo] Mot de passe de leo :
leo n'apparaît pas dans le fichier sudoers. Cet incident sera signalé.
leo@raspberrypi:/home/pi$ exit
exit
pi@raspberrypi:~ $ su maxime
Mot de passe :
maxime@raspberrypi:/home/pi$ sudo useradd test

Nous espérons que vous avez reçu de votre administrateur système local
les consignes traditionnelles. Généralement, elles se concentrent sur ces trois éléments :

    #1) Respectez la vie privée des autres.
    #2) Réfléchissez avant d'utiliser le clavier.
    #3) De grands pouvoirs confèrent de grandes responsabilités.

[sudo] Mot de passe de maxime :
maxime@raspberrypi:/home/pi$ id test
uid=1004(test) gid=1006(test) groupes=1006(test)
maxime@raspberrypi:/home/pi$
```

Source : Auteur

Tout fonctionne comme prévu ! Maxime a bien la possibilité d'utiliser la commande « sudo » et « leo » non.

Détail important, le groupe primaire apparaît dès la création du compte utilisateur, il peut être modifié en groupe secondaire à souhait, il faut comprendre que le groupe primaire est le groupe qui sera assigné par défaut à toute création de ressource sur le système. Donc faire attention à bien réfléchir avant de changer le groupe primaire par défaut de l'utilisateur.

4. Comment gérer les droits d'accès aux fichiers et répertoires ?

Nous allons aborder les droits sous « Linux », toute ressource est appelée objet, que ce soit un répertoire, un fichier ou même un processus. Ils appartiennent à un compte utilisateur ou système ainsi qu'à un groupe d'utilisateur ou système !

Les droits d'accès associés à ces objets sont définis pour par chacun cité précédemment et tous les autres comptes utilisateurs et/ou système qui ne sont pas le propriétaire ni dans le groupe propriétaire. Le propriétaire de l'objet et le super utilisateur peuvent gérer directement les droits d'accès associés à l'objet, ce qui est l'aspect "discrétionnaire" de la gestion de ces droits.

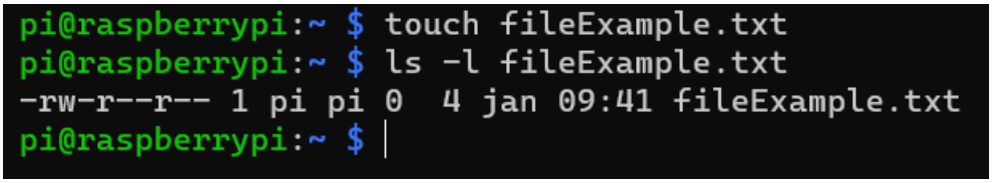
1. La commande « chmod »

Commençons par l'outil « chmod » qui veut dire « change mode », il permet de définir le droit d'accès aux ressources. Il existe trois droits différents :

1. Le droit de lecture symbolisé par « r »
2. Le droit d'écriture symbolisé par « w »
3. Enfin le droit d'exécution symbolisé par « x »

Nous passons directement à la pratique, créons un fichier avec l'utilisateur « pi » et voyons les droits associés, pour créer un fichier, nous utilisons la commande « touch » et pour voir les détails de droits : « ls » avec l'argument « -l » suivi du nom du fichier créé.

Figure 14 - Commande pour créer un fichier et lister ses droits et détails



Source : Auteur

Nous pouvons observer les détails cités précédemment, le « r », « w » et pas de « x ». Tout d'abord, le tiré « - », le plus à gauche peut être soit représenté comme lettre « d » qui signifie « directory » donc répertoire soit le symbole « - » pour fichier. Puis se suivent neuf autres emplacements à droite où il ne peut soit y avoir les trois droits ou le tiré qui représente « droit non affecté ».

Pourquoi neuf emplacements ? Les trois premiers pour l'utilisateur propriétaire de la ressource, les trois seconds pour le groupe propriétaire et le dernier pour tous les autres utilisateurs. Ensuite il est écrit « pi » deux fois, le premier emplacement désigne le nom de l'utilisateur propriétaire et le second, le groupe propriétaire.

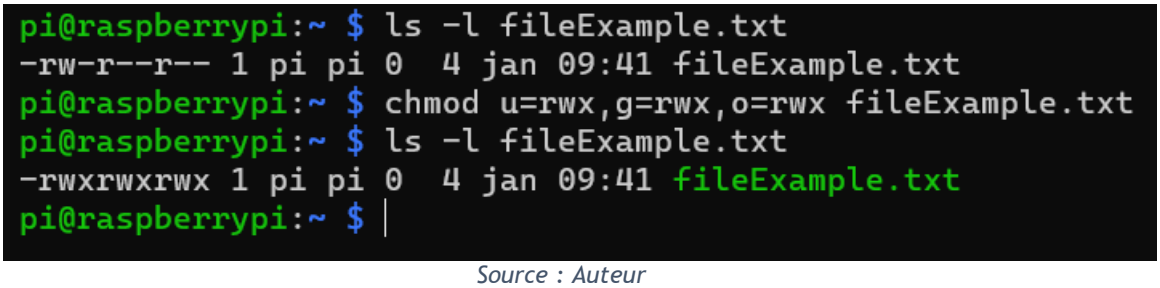
Voici comment lire les droits et propriétaire de cette ligne : « Il s'agit d'un fichier, où l'utilisateur propriétaire nommée « pi » a uniquement le droit de lecture et d'écriture, le groupe propriétaire nomme « pi » a uniquement le droit de lecture et les autres utilisateurs ont le droit de lecture. ».

La représentation des différents utilisateurs est la suivante :

1. Utilisateur propriétaire : « u »
2. Groupe propriétaire : « g »
3. Autre utilisateur : « o »
4. Tous confondus : « a »

Ainsi nous pouvons commencer à utiliser l'outil « chmod » qui permet de modifier le mode d'interaction avec la ressource. Passons par la pratique et nous allons détailler par la suite :

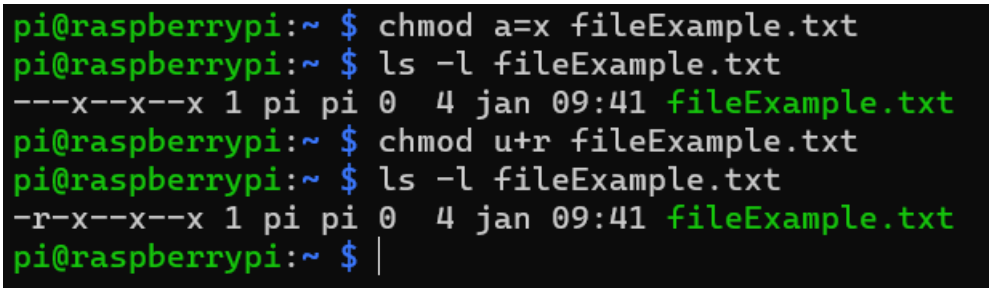
Figure 15 - Commande pour modifier les droits d'un fichier



Source : Auteur

Sur cette figure se trouve la commande « chmod », on remarque qu'en premier argument il y a « u=rwx,g=rwx,o=rwx », nous identifions bien le « u », « g » et « r » et ce qui suit du égal ce sont les droits. Cela veut dire que la commande va changer le mode pour tous les types d'utilisateurs et écraser les anciennes valeurs. Oui, cela va écraser les anciennes valeurs, si nous souhaitons uniquement modifier un utilisateur spécifique et un seul droit voici comment procéder :

Figure 16 - Commande avec options alternatives pour modifier un fichier

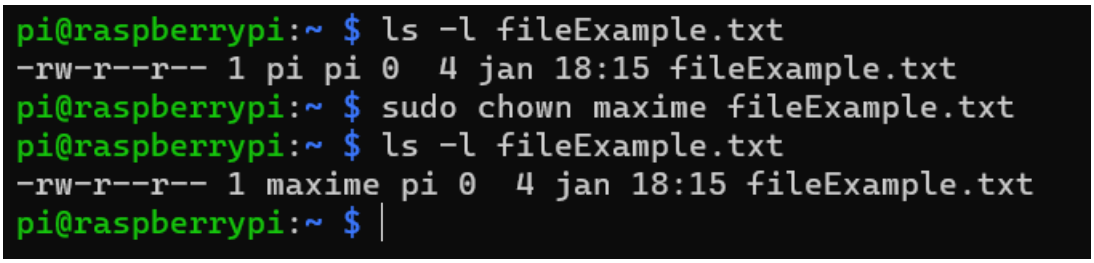


L'utilisation du « + » au lieu du « = » est utile uniquement dans le cas où la valeur de droit que l'on veut modifier est déjà un « - » comme dans l'exemple ci-dessus le moins est aussi possible si le droit est existant : « chmod u-w fileExemple.txt. Petite parenthèse nous pouvons aussi directement associer les types d'utilisateurs entre eux si les droits sont les mêmes, par exemple : « chmod ug=rw fileExample.txt ».

2. La commande « chown »

Nous voyons la commande « chown », qui permet uniquement de changer l'utilisateur propriétaire d'une ressource. Par rapport à la figure ci-dessus, il s'agit de changer l'utilisateur « pi » comme cela :

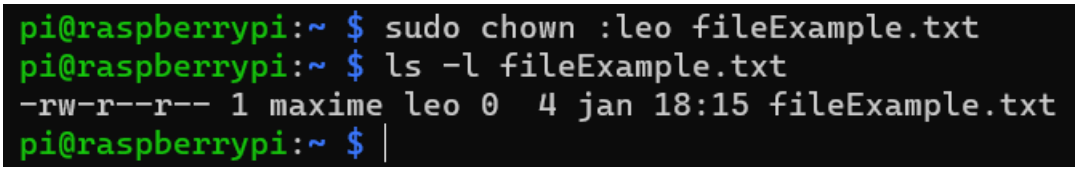
Figure 17 - Commande pour définir le propriétaire de la ressource



Source : Auteur

Nous pouvons aussi changer le groupe propriétaire même s'il y a une commande plus éloquente pour faire cela, nous allons tout de même voir sa syntaxe, il faut ajouter un « : » avant le nom du groupe propriétaire pour préciser qu'il s'agit d'un groupe :

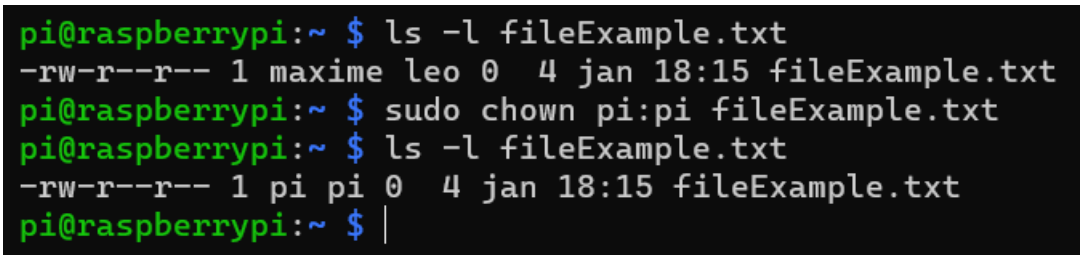
Figure 18 - Commande pour changer uniquement le groupe propriétaire de la ressource



Source : Auteur

Nous pouvons aussi changer l'utilisateur et le groupe propriétaire comme cela :

Figure 19 - Commande pour modifier à la fois l'utilisateur et le groupe propriétaire de la ressource

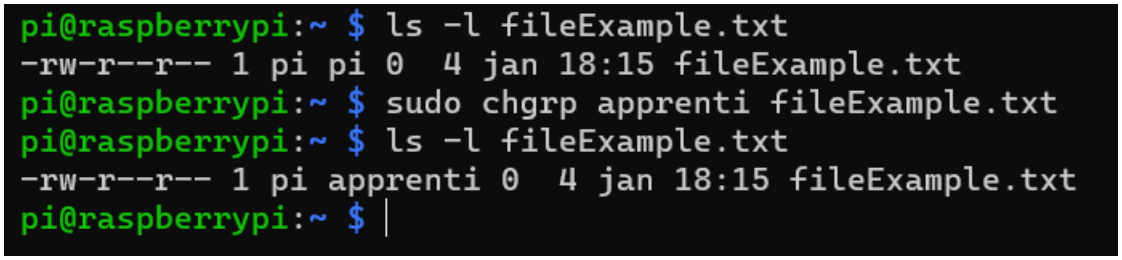


Source : Auteur

3. La commande « chgrp »

Cette commande est plus éloquente mais moins puissante dans le sens de sa diversité d'action possible, elle permet la presque la même chose que la commande précédente mais elle ne peut pas changer d'utilisateur propriétaire ! Voici son utilisation :

Figure 20 - Commande pour modifier le groupe propriétaire de la ressource



Source : Auteur

(On ajoute jack est groupe sec. Dev-senior et on regarde si maxime créé un fichier… Jack peut voir ? Non c'est en groupe secondaire, Donc on peut mettre en groupe primaire dev-senior pour les deux. Parce que c'est le groupe primaire qui définit le groupe propriétaire d'un fichier créé sinon il faudrait faire « chmod -g+rw »)

4. D'autres moyens de sécuriser l'accès aux ressources ?

Nous avons vu le contrôle d'accès par le biais des droits, je propose d'autres solutions existantes pour protéger efficacement l'intégrité de nos données, avec un premier exemple pratique puis des énumérations de solutions possibles :

Une utilisation de logiciel de chiffrement comme l'outil GPG (GNU Privacy Guard) permet de chiffrer et déchiffrer des données numériques en autres avec d'autres capacités comme signer et vérifier des signatures numériques. C'est un algorithme de chiffrement asymétrique, nous pouvons nous référer à l'ancien document « Raspberry Pi – Connexion à distance » où nous avions parlé en détail du « Secure Shell », de son système de clé publique et privée et sa génération ainsi que la comparaison avec le chiffrement symétrique. Nous allons directement installer l'outil en question avec cette commande :

Figure 21 - Commande pour installer GNU Privacy Guard

```
pi@raspberrypi:~ $ sudo apt-get install gnupg
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
gnupg est déjà la version la plus récente (2.2.27-2+deb11u2).
gnupg passé en « installé manuellement ».
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  libfuse2
Veuillez utiliser « sudo apt autoremove » pour le supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 19 non mis à jour.
pi@raspberrypi:~ $
```

Source : Auteur

Nous devons ensuite générer notre clé de chiffrement et suivons les étapes comme sur la figure ci-dessous, il faudra protéger notre clé secrète par une phrase de passe qui permettra, si la clé privée se fait voler, d'être inutilisable :

Figure 22 - Commande pour la génération des clés asymétriques

```
pi@raspberrypi:~ $ gpg --full-generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Sélectionnez le type de clef désiré :
   (1) RSA et RSA (par défaut)
   (2) DSA et Elgamal
   (3) DSA (signature seule)
   (4) RSA (signature seule)
  (14) Existing key from card
Quel est votre choix ? 1
les clefs RSA peuvent faire une taille comprise entre 1024 et 4096 bits.
Quelle taille de clef désirez-vous ? (3072) 1024
La taille demandée est 1024 bits
Veuillez indiquer le temps pendant lequel cette clef devrait être valable.
         0 = la clef n'expire pas
      <n>  = la clef expire dans n jours
      <n>w = la clef expire dans n semaines
      <n>m = la clef expire dans n mois
      <n>y = la clef expire dans n ans
Pendant combien de temps la clef est-elle valable ? (0)
La clef n'expire pas du tout
Est-ce correct ? (o/N) o

GnuPG doit construire une identité pour identifier la clef.

Nom réel : Dasek Joiakim
Adresse électronique : dasek.joiakim@gmail.com
Commentaire :
Vous avez sélectionné cette identité :
    « Dasek Joiakim <dasek.joiakim@gmail.com> »

Changer le (N)om, le (C)ommentaire, l'(A)dresse électronique
ou (O)ui/(Q)uitter ? o
De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
```

Source : Auteur

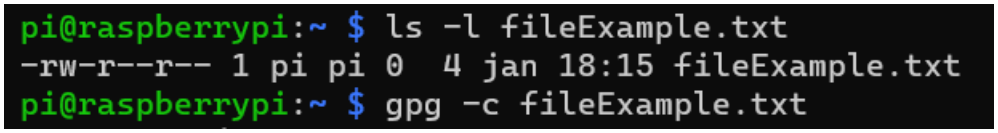
Ensuite on nous demande la phrase de passe :

Figure 23 - Prompt, en attente de validation d'une phrase de passe

Source : Auteur

La génération est faite, ainsi nous avons une clé pour chiffrer et une autre pour déchiffrer ! L'outil de chiffrement est prêt, il nous suffit d'appeler la commande « gpg » suivi de l'option « -c » qui permet de chiffre notre fichier ainsi :

Figure 24 - Commande pour chiffrer un fichier avec GPG



Source : Auteur

Le mot de passe est ensuite demandé et le fichier est chiffré avec l'extension « gpg » :

Figure 25 - Commande pour liste et montrer le fichier chiffré

Source : Auteur

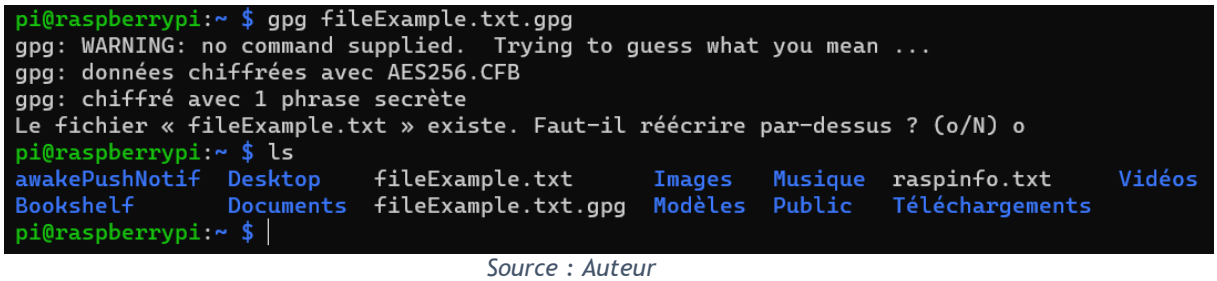
Ainsi nous pouvons déchiffrer le fichier avec extension « gpg » comme cela :

Figure 26 - Commande pour déchiffrer le fichier

Source : Auteur

Le mot de passe et demandé puis le fichier est déchiffré et retrouve l'état initial !

Figure 27 - Résultat de la commande de déchiffrement



Source : Auteur

Nous pouvons voir ici, une liste non exhaustive de différentes manières de protéger des données informatiques. Il faut savoir que ces solutions peuvent être appliquées comme des couches de sécurité allant du plus physique au numérique et peuvent se superposer :

1. Nous pouvons utiliser un système de contrôle d'accès basé sur des rôles (RBAC) pour décider qui a accès à quels fichiers et dossiers. Par exemple, nous pouvons l'utiliser pour donner accès en lecture seule aux fichiers de comptabilité aux comptables et en lecture et écriture aux responsables financiers.
2. Un système de contrôle d'accès contextuel (MAC), pourrait nous permettre de définir des règles qui tiennent compte du contexte qu'il soit interne ou externe, par exemple, nous pourrions définir à ce que le système informatique autorise les lectures de dossier sensibles uniquement pendant les heures prévue à cet effet, en dehors de ce cas, aucun droit accordé !
3. Nous pouvons protéger l'accès à un fichier en demandant un mot de passe à l'utilisateur avant de lui donner accès. Par exemple, avec un système de chiffrement au niveau du disque dur, nous pourrions chiffrer un dossier ou un fichier spécifique de manière à préserver l'information aux personnes autorisées.

Conclusion générale

Nous avons compris les enjeux de la sécurité et de la préservation de l'intégrité des ressources de données ainsi que l'importance de mettre en place un système de contrôle d'accès à ces fichiers, dossiers, processus, etc.

Nous sommes en mesure de créer, modifier et rattacher des utilisateurs à des groupes. En deuxième partie, nous avons compris l'utilité des groupes d'utilisateurs et leurs fonctions liées aux fichiers ou dossiers. Cela nous permet de restreindre ou d'autoriser l'accès à ces ressources en fonction des besoins.

Nous avons pratiqué avec différents scénarios qui pourraient être mis en place dans une entreprise. Ensuite, nous avons étudié une autre manière de restreindre l'accès à une ressource : le chiffrement. Nous avons également examiné une liste d'alternatives aux contrôles d'accès, qui peuvent être utilisées selon les besoins de sécurité de l'entreprise.

Conclusion personnelle

Je pense que comprendre les enjeux de la sécurité et de la préservation de l'intégrité des données est essentiel dans le domaine vaste qu'est l'informatique. Je sais que tous les éléments effectués dans ce document pourront-être du quotidien. C'est très intéressant et cela peut vite devenir complexe si par exemple, l'on inclut plus de personnes dans un projet, donc plus de rôles et responsabilité différentes. Une mise en place d'un système de contrôle élaboré et judicieusement calculé en amont doit être faite !

En outre, j'ai découvert une autre manière de restreindre l'accès à une ressource : le chiffrement. J'ai également étudié une liste d'alternatives aux contrôles d'accès, qui peuvent être utilisées selon les besoins de sécurité de l'entreprise. Je suis convaincu que toutes ces compétences seront très utiles dans ma carrière de formation future !

Références

Linux Command Example - useradd, userdel, usermod, groupadd, last and lastb - LookLinux. (2017, March 4). Consulté le 27 décembre 2022, from http://www.looklinux.com/linux-command-example-useradd-userdel-usermod-groupadd-last-and-lastb/

“adduser/usermod/userdel” - Commands to Manage Users. (n.d.). Consulté le 27 décembre 2022, from https://www.herongyang.com/Linux/User-Group-adduser-usermod-userdel.html

The Complete Guide to User Management in Linux. (2022, Janvier 11). Consulté le 25 décembre 2022, from https://www.makeuseof.com/user-management-linux-guide/

D. I. (n.d.). Gestion des comptes d’utilisateurs sous Linux. Consulté le 26 décembre 2022, from https://www.developpement-informatique.com/article/472/gestion-des-comptes-dutilisateurs-sous-linux

Droits sous Linux : Utilisateurs, Groupes, Permissions - Wiki. (n.d.). Consulté le 28 décembre 2022, from https://www.linuxtricks.fr/wiki/droits-sous-linux-utilisateurs-groupes-permissions

TUTOS.EU : Gérer les groupes sous Linux. (n.d.). Consulté le 29 décembre 2022, from https://www.tutos.eu/1139

L’utilisation de la combinaison des commandes grep et cut | Commandes et Système | IT-Connect. (2012, August 30). Consulté le 2 janvier 2023, from https://www.it-connect.fr/%ef%bb%bflutilisation-de-la-combinaison-des-commandes-grep-et-cut/

cut command in Linux with examples - GeeksforGeeks. (2017, November 17). Consulté le 4 janvier 2023, from https://www.geeksforgeeks.org/cut-command-linux-examples/

Chapter 49. Security and SELinux Red Hat Enterprise Linux 5 | Red Hat Customer Portal. (n.d.). Consulté le janvier 4, 2023, from https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/selg-overview

Le Mandatory Access Control (MAC) : comment fonctionne le contrôle d'accès obligatoire ? (2020, October 14). Consulté le 3 janvier 2023, from https://www.ionos.fr/digitalguide/serveur/securite/quest-ce-que-le-mandatory-access-control-mac/

Role Based Access Control (RBAC) : comment fonctionne le contrôle d'accès à base de rôles ? (2020, October 14). Consulté le 2 janvier 2023, from https://www.ionos.fr/digitalguide/serveur/securite/quest-ce-que-le-role-based-access-control-rbac/

6.8.2. Activer le RBAC (Role-Based Access Control) JBoss Enterprise Application Platform 6.3 | Red Hat Customer Portal. (n.d.). Consulté le 3 janvier 2023, from https://access.redhat.com/documentation/fr-fr/jboss_enterprise_application_platform/6.3/html/security_guide/enabling_role-based_access_control