DOSSIER TECHNIQUE

HARDENING LINUX

Table des matières

Introduction

Le Hardening est un groupe d'action effectué sur une machine Linux permettant d'assurer la sécurité / d'éviter un certain nombre d'actions malveillantes d'une machine ou d'un serveur en déployant une multitude de services.

Le Hardening Système vise à fortifier une machine, un serveur, un poste client, dans l'optique d'en augmenter le niveau de sécurité. Le but premier du hardening est de réduire le nombre d'objets (utilisateurs, bibliothèques, applications, etc.) présents sur le système, en ne conservant que ceux qui sont nécessaires au bon fonctionnement du serveur et du service rendu par ce dernier.

Dans ce guide, nous allons vous présentez avec des exemples comment sécuriser votre serveur Linux.

Chapitre 1. Le Partitionnement

Créer la partition /srv

La première étape de notre guide est de sécuriser notre machine Linux lors de l'installation de celleci. Pour ce faire, nous allons commencer par partitionner notre disque de manière à isoler les services isoler dans une partition différente. En effet, nous allons créer une partition que l'on pourrait nommer /srv qui contiendra tous nos services installés sur notre serveur.

Pour pouvoir installer les services dans la bonne partition, nous vous conseillons de bien partitionner l'outil DPKG qui vous permettra d'installer les paquets des services au bon endroit.

Cette partition nous servira pour y mettre tous les services relatifs à notre serveur Linux. Cela nous sera utile pour sauvegarder nos services lors d'un backup. En effet, nous devrons seulement sélectionner le dossier /srv si nous voulons sauvegarder seulement nos services installés sur notre serveur.

Nous vous avons mis en annexe un lien qui explique comment créer une nouvelle partition.

Chapitre 2. Mot de passe root

Introduction

L'utilisateur root est un utilisateur qui aura les accès maximums pour notre serveur Linux. Pour garantir une sécurité optimale sur notre serveur, le but est de restreindre au maximum l'accès à cet utilisateur pour éviter que quelqu'un puisse prendre le contrôle de cet utilisateur et qui puisse accéder aux services de notre machine.

Création du mot de passe

La première des choses à faire est de choisir un mot de passe complet lors de l'installation de notre OS. Un mot de passe dit « complet » doit contenir au minimum 12 signes mélangeant des majuscules, des minuscules, des chiffres et des caractères spéciaux.

La deuxième chose à faire est de restreindre l'accès à l'utilisateur root même aux administrateurs. Pour pallier ça, l'administrateur devra utiliser sudo sur sa session.

Nous vous avons mis en annexe un lien qui explique comment créer un mot de passe root.

Chapitre 3. Gestion des comptes utilisateurs

Introduction

Pour être encore plus précis sur l'utilisation des utilisateurs de notre serveur Linux, nous allons voir comment optimiser l'accès à notre machine. La première étape est de supprimer les comptes utilisateur qui ne sont pas utilisés.

Installation

Pour faire cela, nous allons en sudo utiliser la commande : usermod -L

Cette commande permet de verrouiller un utilisateur et donc de le rendre inutilisable.

La deuxième étape serait de restreindre le temps d'accès d'une session à distance pour éviter tout vol de session ou encore de connexions par des personnes qui ne devraient pas se connecter.

Ce qui pourrait être encore plus performant serait d'installer un gestionnaire de mot de passe qui pourra gérer les mots de passe pour tous les utilisateurs présents sur notre serveur.

Une autre manière de sécuriser un compte serait d'activer la double authentification à chaque fois que cela est possible.

Nous vous avons mis en annexe un détail de la commande Usermod -L.

Chapitre 4. Sécuriser SSH

Introduction

Pour que nos utilisateurs puissent se connecter au serveur distant, nous utiliserons le protocole SSH qui est , en plus d'être sécurisé, est très performant.

Pour commencer à sécuriser ce protocole, nous allons en premier lieu changer le port par défaut de SSH qui est le port 22. C'est sur ce port que toutes les attaques de connexions à notre serveur seront faites.

Installation

Pour éditer le fichier de configuration du port SSH, nous utiliserons la commande :

```
nano /etc/ssh/sshd_config
```

Il faut rechercher la ligne appropriée et remplacer le port 22 par un numéro de votre choix. Cependant, il est nécessaire de prendre en compte le fait qu'il existe plusieurs ports standards pour d'autres services (comme le port 80 pour HTTP) que vous ne devriez pas utiliser si possible pour cette raison. Ainsi, il est recommandé de jeter d'abord un coup d'œil à la liste des ports logiciels (RCP et UDP).

La seconde étape est de désactiver l'accès SSH à l'utilisateur root. L'accès à l'utilisateur root en SSH est fortement déconseillé. C'est en effet l'utilisateur qui possède tous les droits sur le système, donc si son accès SSH est compromis, tout votre serveur sera compromis. Pour faire cela, il faut se rendre dans le fichier de config suivant `/etc/ssh/sshd_config.` Puis, nous avons juste à passer la ligne PermitRootLogin no. Une fois cela fait, l'utilisateur root n'a plus accès à SSH.

Maintenant, nous allons préciser quels utilisateurs de notre serveur auront accès à SSH. Toujours dans le même fichier, nous devons rentrer sur une ligne vierge :

AllowUsers

Exemple :

```
AllowUsers *@10.* pi@192.168.1.* vincent@192.168.1.56
```

Pour appliquer toutes les modifications, veuillez redémarrer le service SSH : sudo service ssh restart

Nous vous avons mis en annexe comment configurer en détail ssh sur un serveur Linux.

Chapitre 5. Sécuriser le serveur des attaques Brute Force

Introduction

Pour éviter que notre serveur Linux soit victime de pirates qui utilisent des attaques telles que des attaques Brute Force, nous allons utiliser un outil qui se nomme Fail2Ban. Cet outil est un logiciel libre de sécurité développé en langage Python (littéralement « un échec de tentative entraîne une interdiction ») est un module de serveur Web que l'on peut utiliser sur les systèmes Linux et POSIX accompagnés soit, de pare-feu soit de filtres de paquets. Fail2ban communique dans les fichiers de logs les adresses IP potentiellement suspectes : par exemple, au cas où ces adresses essayent à plusieurs reprises de s'identifier sur un compte email avec de mauvais identifiants. Un certain nombre de tentatives échouées entraîne ensuite automatiquement le blocage de l'adresse IP pendant une période préalablement programmée. Par ailleurs, l'administrateur a la possibilité de configurer Fail2ban de sorte que chaque adresse IP suspecte lui soit communiquée par email.

Par défaut, un Fail2ban contient toute une gamme de filtres tels que pour Apache, Postfix ou Courier Mail Server qui peuvent être reconnues grâce à une séquence particulière de caractères dans les fichiers de logs. Les filtres peuvent enclencher des actions : il s'agit de commandes paramétrées pour une période définie. La combinaison d'un filtre et d'une action, par exemple le blocage d'une adresse IP grâce à Fail2ban, est nommée prison (jail). Il est possible de programmer ces prisons avec Fail2ban

via n'importe quel logiciel pouvant créer des fichiers de logs. Etant donné que Fail2ban est un framework open source bénéficiant de la licence GPL2 (Licence publique générale), il peut être utilisé et étendu gratuitement.

Installation

Pour installer ce logiciel, nous allons utiliser la commande : sudo aptget

install fail2ban

Une fois que cela est fait, nous allons démarrer ce service en utilisant la commande :

sudo service fail2ban

Voilà, le framework est à présent installé, nous allons maintenant vous montrer comment

le configurer pour votre serveur Exemple :

```
[DEFAULT]
ignoreip = 127.0.0.1/8 192.168.1.100/24
bantime  = 600
findtime  = 600
maxretry = 3
destemail = looklinux@gmail.com
```

Ignoreip permet de whitelist les IPs autorisées à la connexion.

bantime est le temps de bannissement d'une IP non autorisée qui essaye de se connecter sur le réseau. maxretry est le nombre d'essais maximum que peut utiliser une ip pour essayer de se

connecter à

Vous retrouverez en annexe un tutoriel pour installer et configurer Fail2ban correctement.

Chapitre 6. Le Pare-Feu

Introduction

La configuration d'un pare-feu est quelque chose de très important pour bloquer les attaques extérieures sur votre serveur. Heureusement, il existe quelques outils pour vous faciliter la tâche !

Installation

Pour cela, nous allons utiliser le pare-feu UFW. Voici la commande pour l'installer :

```
sudo apt-get install ufw
```

Une fois celui-ci installer, nous allons pouvoir maintenant le configurer :

Pour autoriser l'accès à un port précis en tcp :

sudo ufw allow X/tcp

Pour autoriser l'accès à un port précis en udp :

sudo ufw allow X/udp

Pour refuser l'accès à un port précis en tcp :

sudo ufw deny X/tcp

Pour refuser l'accès à un port précis en udp :

sudo ufw deny X/udp

Ici, X correspond au port que vous souhaitez autoriser / refuser

Maintenant, nous allons voir comment autoriser les connexions d'adresses ip sur le pare-feu :

Pour autoriser une adresse précise :

sudo ufw allow from 192.168.1.3 Pour

refuser une adresse précise : sudo ufw

deny from 192.168.1.3

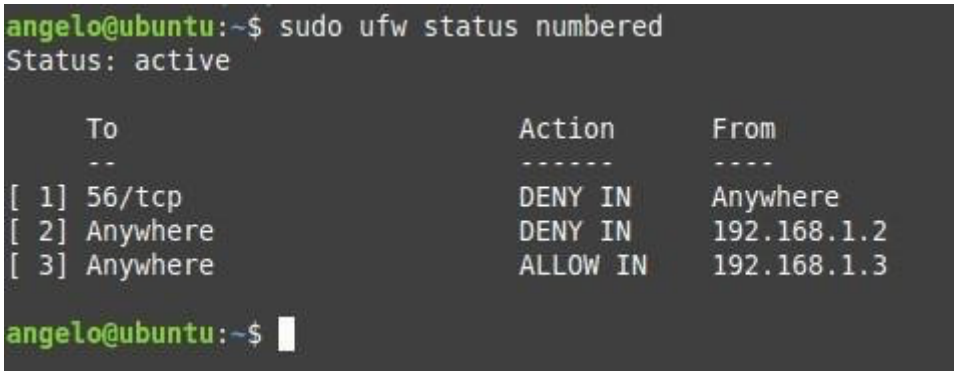
Ou sur un port précis : sudo ufw allow from

192.168.1.3 to any port X pour refuser :

sudo ufw deny from 192.168.1.3 to any port X

Pour énumérer toutes les règles de notre pare-feu, nous allons utiliser la commande :

sudo ufw status numbered



Pour supprimer une règle précise, nous pouvons utiliser la commande :

sudo ufw delete 3

Grâce à toutes ces commandes, vous savez maintenant comment ajouter des règles, en supprimer ou encore autoriser l'écoute de certains ports ou justement la refuser.

Vous retrouverez en annexe comment configurer correctement UFW.

Chapitre 7. Backup

Introduction

Cette section vous montrera comment sauvegarder l'intégralité de notre système linux. La sauvegarde est une étape importante de ce guide, car en cas d'attaque tel que des ransomware ou tout autre type d'attaques qui pourraient potentiellement supprimer, bloquer ou encore détruire des fichiers importants sur notre serveur, nous aurions la capacité de sauvegarder tous nos fichiers et ainsi pouvoir les redéployer sur un autre serveur.

Installation

Pour sauvegarder l'ensemble du système, il vous suffit d'ouvrir votre terminal et d'exécuter la commande suivante en tant qu'utilisateur root :

$ sudo rsync -aAXv / --exclude = {"/ dev / *", "/ proc / *", "/ sys / *", "/ tmp / *", "/ run / *", "/ mnt / * "," / media / * "," / lost + found "} / mnt

Cette commande sauvegardera tout le répertoire racine (/), à l'exclusion des répertoires / dev, / proc, / sys, / tmp, / run, / mnt, / media, / lost + found, et enregistrera les données dans le dossier / mnt.

Nous avons donc dans le dossier /mnt toute la sauvegarde de notre serveur Linux.

Nous ne précisons pas d'exclure notre dossier /srv car c'est celui-ci qui contient tous nos services présents sur notre serveur.

Vous trouverez en annexe le lien vers un script permettant à l'exécution de sauvegarder tout votre système.

Vous retrouverez en annexe comment cloner ou sauvegarder une partition sur Linux.

Chapitre 8. Lynis

Introduction

Lynis est un script shell permettant d'auditer votre environnement Linux de fond en comble, de vous donner un indice de sécurisation de votre environnement et des recommandations pour améliorer la sécurité de votre environnement.

Les objectifs de Lynis sont simples :

Réaliser des tests automatiques de sécurité.

Tester la compliance à des normes de sécurité (ISO27001, PCI-DSS, HIPAA).

Détecter des vulnérabilités.

Lynis peut aussi vous assister dans la vérification de votre configuration de Hardening, tester une escalade de privilèges, vérifier le bon patching de votre serveur, détecter une intrusion... Nous allons voir à présent comment l'installer

Installation

Pour installer Lynis, nous allons d'abord installer toutes les dépendances pour que ce logiciel fonctionne :

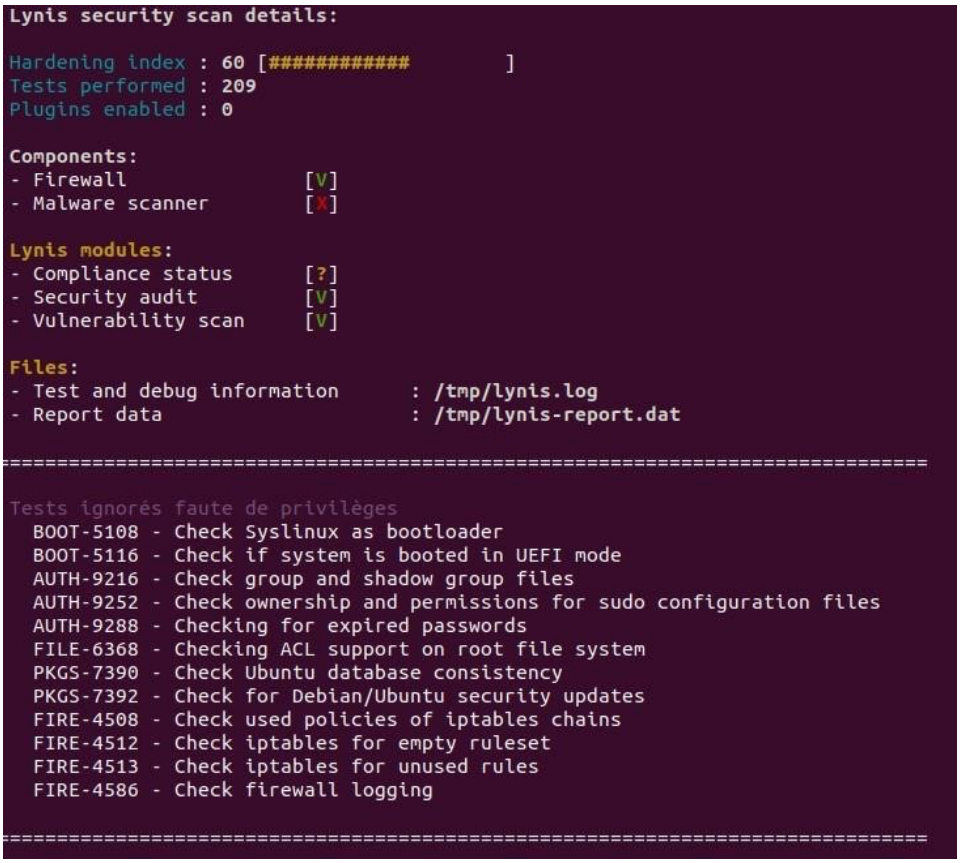
sudo wget -O – https://packages.cisofy.com/keys/cisofy-software-public.key | sudo apt-key add –

sudo apt install apt-transport-https

echo "deb https://packages.cisofy.com/community/lynis/deb/ stable main" | sudo tee /etc/apt/sources.list.d/cisofylynis.list

sudo apt update sudo apt install lynis

lynis audit system Exemples :



Si vous souhaitez personnaliser vos checks car vous ne souhaitez pas tout vérifier ou simplement parce que vous avez développé un plugin, vous avez la possibilité de modifier le fichier de configuration. vi /etc/lynis/default.prf

Si vous souhaitez par exemple désactiver un plugin, vous devez commenter la ligne correspondant à celui-ci.

Vous trouverez en annexe comment installer et configurer Lynis.

Chapitre 9. Cron

Introduction

Crontab est utile pour effectuer diverses opérations telles que la gestion de la sauvegarde automatisée, la rotation des fichiers journaux, la synchronisation des fichiers entre des machines distantes et la suppression des dossiers temporaires, etc. Crontab peut être utilisé pour tout type de travail, mais il devient particulièrement utile lorsque nous commençons à traiter l'administration du système type de travail.

En utilisant cron, un administrateur peut planifier une tâche à exécuter à une heure et un jour spécifiques. Cela peut nous être utile pour pouvoir automatiser des taches de sécurité sur notre serveur Linux.

Voyons donc comment nous pouvons mettre cela en place.

Installation

Pour utiliser cron, une syntaxe précise est requise pour automatiser certaines taches :

La syntaxe est la suivante : mm hh jj MMM JJJ user commande

Voici un exemple de commande que l'on veut automatiser :

01 * * * * root echo "cette commande est exécutée toutes les heures passées d'une minute"

17 8 * * * root echo "Cette commande est exécutée tous les jours à 08h17"

17 20 * * * root echo "Cette commande est exécutée tous les jours à 20h17"

00 4 * * 0 root echo "Cette commande est exécutée tous les dimanches à 4h00"

42 4 1 * * root echo "Cette commande est exécutée tous les 1ers du mois à 4h42"

01 * 19 07 * root echo "Cette commande est exécutée toutes les heures passées d'une minute tous les 19 Juillet"

Nous pouvons donc maintenant automatiser des taches que nous souhaitons répéter comme le redémarrage d'un serveur web apache tous les jours à 12h00 avec la commande :

00 12 * * * sudo services restart apache2

Avec cette commande, tous les jours à 12H00 le service apache sera redémarrer.

Nous vous avons mis en annexe un lien qui explique comment automatiser des taches avec cron.

Chapitre 10. Chiffrement

Introduction

Notre serveur Linux contient des données sensibles que nous ne pouvons pas négliger et que nous sommes obligées de sécuriser pour éviter toute utilisation de la part d'utilisateurs non apte à lire, modifier ou encore exécuter ces fichiers. Nous utiliserons donc OpenSSL qui nous permettra de chiffrer un fichier ou un dossier avec un mot de passe définit par l'administrateur sur chaque fichier ou dossier.

Installation

Pour commencer, nous allons installer OpenSSL sur notre serveur :

apt-get install openssl

OpenSSL s'utilise avec le format :

```
openssl <comande> [options]
```

Pour voir notre version de OpenSSL, nous utilisons la commande :

openssl version

Pour commencer à encrypter un fichier, nous utiliserons cette commande :

```
commande = $(openssl enc -e -aes-256-cbc -in $file -out $filefin)
```

-aes-256-cbc : C'est le type d'encodage que nous souhaitons utiliser, il en existe plein d'autres.

$file : C'est le nom de notre fichier que l'on souhaite encrypter

$filefin : C'est le nom de notre fichier final qui sera encrypter par un mot de passe que nous aurons défini.

Maintenant que nous savons comment chiffrer n'importe quel fichier, nous pouvons le mettre en place sur notre serveur pour chiffrer des fichiers de configurations ou encore des fichiers contenants des mots de passe qui doivent rester secrets.

Vous retrouverez en annexe un tutoriel détaillé pour chiffrer correctement ses fichiers / dossiers.

Chapitre 11. Mise à jour automatique

Introduction

La mise à jour d'un système linux ainsi que ses paquets est une étape importante de la sécurisation de notre serveur. En effet, la mise à jour permet d'avoir les dernières versions des outils qui sont installés sur notre serveur. Les mises à jour servent aussi à maintenir la stabilité de notre serveur et d'avoir les dernières versions de sécurité pour chaque outil. Un outil obsolète peut être une faille de sécurité sur notre serveur et donc peut permettre à de potentiels pirates de rentrer plus facilement dans notre infrastructure. Ces mises à jour peuvent être automatisées avec Cron que nous vous avons présenté dans le chapitre 9.

Installation

Pour mettre à jour nos serveurs, il y a plusieurs étapes à effectuer :

La première étape est de mettre à jour la liste de nos paquets :

sudo apt update

La deuxième étape est de lister tous les paquets de notre liste pouvant être mis à jour :

sudo apt list –upgradable

La troisième étape permet d'exécuter la mise à jour. Elle est divisée en deux options :

- La mise à jour des paquets
- La mise à jour des paquets et du système pour faire la mise à jour des paquets

:

sudo apt upgrade

Pour faire la mise à jour des paquets et du système :

sudo apt full-upgrade

Il est donc possible de cumuler ces commandes et de faire un script bash qui met à jour notre système et nos paquets. Il est aussi possible de l'intégrer dans un fichier Cron pour exécuter le script à des heures et jours spécifiques pour avoir un système à jour même quand il n'y a personne sur la machine. Par exemple, pour une entreprise, la mise à jour pourrait être prévue avec Cron la nuit pour éviter de déranger les services en cours la journée.

Nous vous avons mis en annexe un lien qui explique comment mettre à jour automatiquement.

Chapitre 12. Configurer une MFA

Introduction

Pour améliorer notre sécurité sur notre serveur, nous pouvons mettre en place l'authentification à deux facteurs (MFA). Cela permet à un utilisateur qui tente de se connecter sur notre serveur de devoir rentrer un mot de passe, mais aussi un jeton secret. La MFA peut être couplé avec SSH pour avoir une double authentification avec SSH. Ensemble, ces éléments rendent le serveur plus résistant aux tentatives de connexion non autorisées par force brute et peuvent améliorer la sécurité du cloud pour les petites entreprises.

Installation

Plusieurs étapes sont requises pour pouvoir mettre en place ce système de double authentification :

La première étape est d'installer la librairie en question :

```
sudo apt-get install libpam-google-authenticator
```

La deuxième étape est d'installer les prérequis pour cette librairie :

```
sudo apt-get install wget make gcc libpam0g-dev
```

La troisième étape est de lancer l'installation, normalement un dossier a été créé. On se place dedans et on exécute la commande :

```
sudo make install
```

Voilà, l'outil est maintenant installé, il vous reste plus qu'à le configurer en fonction de votre utilisation.

Nous vous avons mis en annexe un lien qui explique comment configurer une MFA.

Chapitre 13. Mots de passe utilisateurs

Introduction

Après avoir vu comment renforcer la sécurité de notre serveur en améliorant le mot de passe de l'utilisateur root, nous allons voir comment renforcer la sécurité, mais cette fois avec les mots de passe des utilisateurs de notre serveur. Pour améliorer la sécurité, nous pouvons mettre en place un système qui oblige un utilisateur à changer son mot de passe tous les x temps. Cette pratique permet de ne jamais garder les mêmes mots de passe et donc d'augmenter la difficulté d'intrusion sur notre serveur. Pour faire notre rotation régulière des mots de passe, nous allons utiliser un outil qui s'appelle Chage.

Installation

Installation de la commande :

sudo apt-get install chage

Vérifier les informations de rotation d'un utilisateur :

```
sudo chage -l user
```

Changer la date de rotation de mot de passe :

```
sudo chage -d YYYY-MM-DD user
```

Changer la date d'expiration d'un compte :

```
sudo chage -E YYYY-MM-DD user
```

Changer le temps minimum de changer son mot de passe pour un utilisateur :

```
sudo chage -m NUM_DAYS user
```

Changer le laps de temps du message d'avertissement qu'un utilisateur peut recevoir pour changer son mot de passe :

```
sudo chage -W NUM_DAYS user
```

Mettre que le mot de passe ne change jamais pour les utilisateurs :

```
sudo chage -M -1 user
```

Maintenant que vous avez configuré chage, vous pouvez à présent l'utiliser correctement pour votre serveur Linux.

Nous vous avons mis en annexe un lien qui explique comment gérer la rotation des mots de passe utilisateurs.

Annexe :

Partitions : Comment créer une partition d'installation Linux ? (lojiciels.com)

Usermod : La commande usermod : exemples et utilisations - malekal.com

SSH : Comment configurer l'authentification par clé SSH sous Linux pour des connexions plus sûres ? - ZDNet

Fail2ban: How to Install and Configure Fail2Ban to Secure Linux Server (linuxhandbook.com)

UFW : How to Configure the UFW Firewall in Linux - Appuals.com

Backup : Comment cloner une partition ou un disque dur sous Linux - JN Community (jaguarnetwork.com)

Lynis : How to Install and Run Lynis on Ubuntu Linux (linoxide.com)

Chiffrement : https://angristan.fr/chiffrer-fichier-openssl-linux/

Apt-get : http://www.octetmalin.net/linux/tutoriels/apt-get.php

Cron : https://www.linuxtricks.fr/wiki/cron-et-crontab-le-planificateur-de-taches

Chage : chage command in Linux with examples - GeeksforGeeks

MFA : https://www.it-connect.fr/linux-comment-activer-le-mfa-sur-un-acces-ssh/

Mot de passe root : https://www.linuxtricks.fr/wiki/ubuntu-initialiser-le-mot-de-passe-root