

Systeme, Scripts et Sécurité

Job 02 Commandes de recherche avancée :

```
dome@debianLaPlateformeSSS:~$ ls
Bureau Documents Images Modèles Musique Public Téléchargements Vidéos
dome@debianLaPlateformeSSS:~$ cd Bureau/
dome@debianLaPlateformeSSS:~/Bureau$ sudo nano mon_texte.txt
[sudo] Mot de passe de dome :
dome@debianLaPlateformeSSS:~/Bureau$ cd
dome@debianLaPlateformeSSS:~$ cd Documents/
dome@debianLaPlateformeSSS:~/Documents$ sudo nano mon_texte.txt
dome@debianLaPlateformeSSS:~/Documents$ cd
dome@debianLaPlateformeSSS:~$ cd Téléchargements/
dome@debianLaPlateformeSSS:~/Téléchargements$ sudo nano mon_texte.txt
dome@debianLaPlateformeSSS:~/Téléchargements$ cd
dome@debianLaPlateformeSSS:~$ cd Vidéos/
dome@debianLaPlateformeSSS:~/Vidéos$ sudo nano mon_texte.txt
dome@debianLaPlateformeSSS:~/Vidéos$ cd
dome@debianLaPlateformeSSS:~$ cd Images/
dome@debianLaPlateformeSSS:~/Images$ sudo nano mon_texte.txt
dome@debianLaPlateformeSSS:~/Images$
```

Pour localiser les cinq fichiers "mon_texte.txt"

```
sudo find / -type f -name "mon_texte.txt" 2>/dev/null
```

Cette commande va rechercher de manière récursive à partir de la racine ("/") tous les fichiers portant le nom "mon_texte.txt" et afficher leur chemin absolu. L'option "-type f" spécifie que seuls les fichiers seront recherchés, et "2>/dev/null" permet de rediriger les erreurs vers /dev/null pour éviter les messages d'erreur.

```
dome@debianLaPlateformeSSS:~$ sudo find / -type f -name "mon_texte.txt" 2>/dev/null
/home/dome/Bureau/mon_texte.txt
/home/dome/Téléchargements/mon_texte.txt
/home/dome/Vidéos/mon_texte.txt
/home/dome/Images/mon_texte.txt
/home/dome/Documents/mon_texte.txt
dome@debianLaPlateformeSSS:~$
```

Job 03 Compression et décompression de fichiers :

- Créez un répertoire nommé "Plateforme" dans le dossier "Documents"

```
dome@debianLaPlateformeSSS:~$ mkdir ~/Documents/Plateforme
dome@debianLaPlateformeSSS:~$ cd Documents/
dome@debianLaPlateformeSSS:~/Documents$ ls
mon_texte.txt  Plateforme
```

- Ajouter le fichier "mon_texte.txt" précédemment créé.

```
dome@debianLaPlateformeSSS:~$ cd Documents/Plateforme/
dome@debianLaPlateformeSSS:~/Documents/Plateforme$ ls
mon_texte.txt
dome@debianLaPlateformeSSS:~/Documents/Plateforme$
```

- Dupliquer "mon_texte.txt" quatre fois dans le même répertoire, formant ainsi un total de cinq fichiers dans le répertoire "Plateforme".

```
dome@debianLaPlateformeSSS:~/Documents/Plateforme$ cp mon_texte.txt mon_texte_2.txt && cp mon_texte.txt mon_texte_3.txt && cp m
on_texte.txt mon_texte_4.txt && cp mon_texte.txt mon_texte_5.txt
dome@debianLaPlateformeSSS:~/Documents/Plateforme$ ls
mon_texte_2.txt mon_texte_3.txt mon_texte_4.txt mon_texte_5.txt mon_texte.txt
dome@debianLaPlateformeSSS:~/Documents/Plateforme$
```

- Archivez le répertoire "Plateforme" en utilisant les commandes "tar" et "gzip".

```
dome@debianLaPlateformeSSS:~/Documents$ tar -cvf Plateforme.tar Plateforme
Plateforme/
Plateforme/mon_texte_4.txt
Plateforme/mon_texte_5.txt
Plateforme/mon_texte.txt
Plateforme/mon_texte_2.txt
Plateforme/mon_texte_3.txt
dome@debianLaPlateformeSSS:~/Documents$ ls
Plateforme  Plateforme.tar
dome@debianLaPlateformeSSS:~/Documents$
```

« Ici l'utilisations de la commande **tar -cvf** »

- *Le format .tar crée une archive regroupant plusieurs fichiers sans les compresser individuellement, nécessitant l'utilisation d'un autre format de compression comme .gz (gzip) pour compresser l'ensemble des fichiers en un seul fichier/archive*

Explication des options utilisées :

- **tar** : Commande pour manipuler des archives.
- **c** : Crée une nouvelle archive.
- **v** : Mode verbeux, affiche les fichiers compressés en détail.
- **f** : Indique le nom du fichier d'archive à créer.

```
dome@debianLaPlateformeSSS:~/Documents$ ls
Plateforme
dome@debianLaPlateformeSSS:~/Documents$ sudo tar -czvf Plateforme.tar.gz Plateforme/
Plateforme/
Plateforme/mon_texte_4.txt
Plateforme/mon_texte_5.txt
Plateforme/mon_texte.txt
Plateforme/Plateforme.tar.gz
Plateforme/mon_texte_2.txt
Plateforme/mon_texte_3.txt
dome@debianLaPlateformeSSS:~/Documents$ ls
Plateforme  Plateforme.tar.gz
dome@debianLaPlateformeSSS:~/Documents$
```

« Ici l'utilisations de la commande tar-gzip = **gzip** »

gzip : Compression rapide avec efficacité modérée, extension ".gz".

Les paramètres utilisés dans la commande **tar -czvf** est la suivante :

- **-c** : Crée une nouvelle archive.
- **-z** : Cette option indique à tar d'utiliser gzip pour compresser l'archive.
- **-v** : Cette option affiche les fichiers pendant qu'ils sont traités.
- **-f** : Cette option indique le nom du fichier d'archive.

```

Plateforme/mon_texte_3.txt
dome@debianLaPlateformeSSS:~/Documents$ ls
Plateforme  Plateforme.tar.gz
dome@debianLaPlateformeSSS:~/Documents$ tar -cjvf Plateforme.tar.bz2 Plateforme/
Plateforme/
Plateforme/mon_texte_4.txt
Plateforme/mon_texte_5.txt
Plateforme/mon_texte.txt
Plateforme/Plateforme.tar.gz
Plateforme/mon_texte_2.txt
Plateforme/mon_texte_3.txt
dome@debianLaPlateformeSSS:~/Documents$ ls
Plateforme  Plateforme.tar.bz2  Plateforme.tar.gz
dome@debianLaPlateformeSSS:~/Documents$

```

« Ici l'utilisations de la commande « tar- bzip2 » = **bzip2** »

***bzip2** : Meilleure efficacité que gzip mais plus lent, extension ".bz2"*

Les paramètres utilisés dans la commande **tar -cjvf** ont les significations suivantes :

- **-c** : Cette option crée une nouvelle archive.
- **-j** : Cette option spécifie à tar d'utiliser bzip2 pour compresser l'archive.
- **-v** : Cette option affiche les fichiers pendant qu'ils sont traités.
- **-f** : Cette option indique le nom du fichier d'archive.

Autres options de compression existent, voici un résumé :

- **xz** : Meilleure compression que gzip et bzip2, plus lent que gzip mais plus rapide que bzip2. extension ".xz".
- **zip** : Compression de collections de fichiers en un seul fichier, compatible avec divers systèmes d'exploitation.

- Décompressez les archives créées en utilisant les commandes appropriées. Explorez les diverses options de décompression.

Pour extraire les fichiers d'une archive tar, utilisez la commande **tar -xf** suivie du nom de l'archive à extraire. Par exemple :

tar -xf exemple.tar

Pour décompresser des fichiers **.tar.gz**, vous pouvez utiliser la commande :

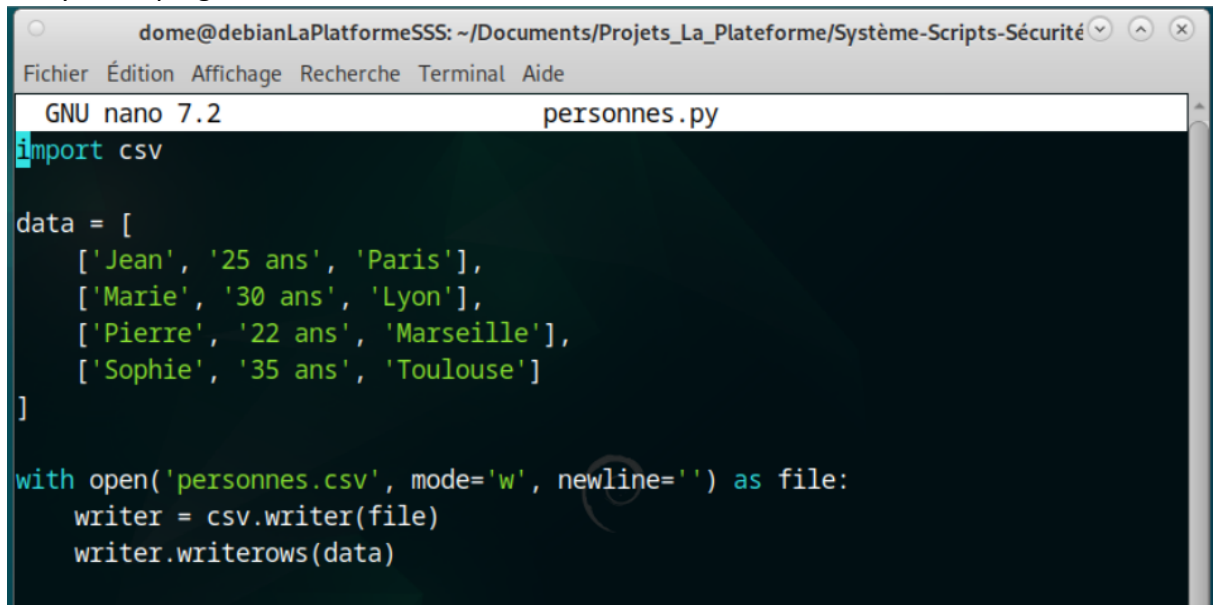
gzip -d filename.tar.

Job 04 Manipulation de texte

C'est quoi un fichier CSV?

« Un fichier CSV (Comma-Separated Values) est un type de fichier texte utilisé pour stocker des données tabulaires sous forme de valeurs séparées par des virgules. Il est largement utilisé pour l'échange de données entre différentes applications, telles que les feuilles de calcul et les bases de données. Les fichiers CSV peuvent être ouverts et modifiés avec des éditeurs de texte simples ou des logiciels de tableur comme Microsoft Excel ou Google Sheets »

Ici Python qui génère un fichier CSV avec les données fournies :

A screenshot of a terminal window titled 'dome@debianLaPlateformeSSS: ~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité'. The window shows the GNU nano 7.2 editor editing a file named 'personnes.py'. The code in the file is as follows:

```
import csv

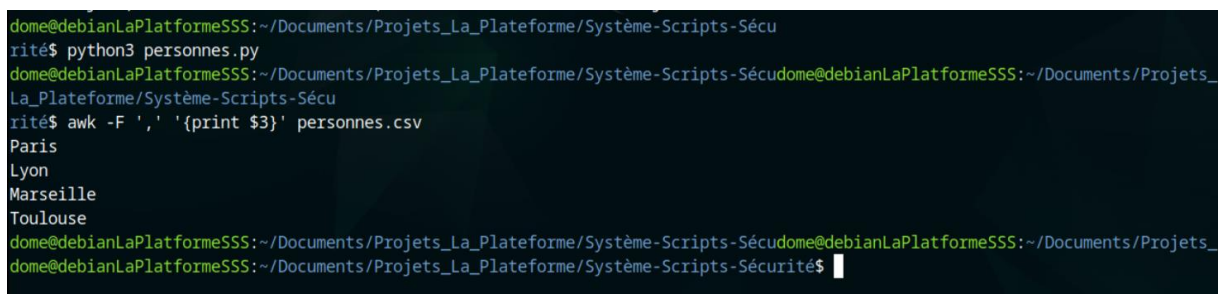
data = [
    ['Jean', '25 ans', 'Paris'],
    ['Marie', '30 ans', 'Lyon'],
    ['Pierre', '22 ans', 'Marseille'],
    ['Sophie', '35 ans', 'Toulouse']
]

with open('personnes.csv', mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerows(data)
```

Une fois exécuté ce script pour créer le fichier CSV "personnes.csv", j'utilise la commande "awk" pour extraire les informations relatives aux villes de chaque personne :

awk -F ',' '{print \$3}' personnes.csv

Dans cette commande "awk" utilise comme délimiteur la virgule (','), puis elle affiche la troisième colonne de chaque ligne du fichier CSV, qui correspond aux villes des personnes.

A screenshot of a terminal window showing the execution of a Python script and an awk command. The terminal output is as follows:

```
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ python3 personnes.py
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ awk -F ',' '{print $3}' personnes.csv
Paris
Lyon
Marseille
Toulouse
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$
```


« L'option "-f" dans la commande "awk" est utilisée pour spécifier un fichier de script AWK à exécuter. Lorsque vous utilisez l'option "-f" suivie du nom d'un fichier, "awk" exécute les instructions contenues dans ce fichier plutôt que d'écrire directement les instructions dans la ligne de commande. »

Par exemple, si vous avez un script AWK complexe que vous souhaitez exécuter sur un fichier CSV, vous pouvez le sauvegarder dans un fichier nommé "mon_script.awk" et l'exécuter avec la commande suivante :

awk -f mon_script.awk fichier.csv

Cela permet d'exécuter les instructions contenues dans "mon_script.awk" sur le fichier "fichier.csv". L'utilisation de l'option "-f" rend le processus plus modulaire et facilite la maintenance et la réutilisation des scripts AWK.

Job 05 Gestion des processus

Pour recenser tous les processus actifs sur notre système, utiliser la commande suivante :

ps aux

```
dome@debianLaPlateformeSSS:~$ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.6	167764	12296	?	Ss	17:25	0:00	/sbin/init
root	2	0.0	0.0	0	0	?	S	17:25	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	17:25	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	17:25	0:00	[rcu_par_gp]
root	5	0.0	0.0	0	0	?	I<	17:25	0:00	[slub_flushwq]
root	6	0.0	0.0	0	0	?	I<	17:25	0:00	[netns]
root	10	0.0	0.0	0	0	?	I<	17:25	0:00	[mm_percpu_wq]
root	11	0.0	0.0	0	0	?	I	17:25	0:00	[rcu_tasks_kthread]
root	12	0.0	0.0	0	0	?	I	17:25	0:00	[rcu_tasks_rude_kthread]
root	13	0.0	0.0	0	0	?	I	17:25	0:00	[rcu_tasks_trace_kthread]
root	14	0.0	0.0	0	0	?	S	17:25	0:00	[ksoftirqd/0]
root	15	0.0	0.0	0	0	?	I	17:25	0:00	[rcu_preempt]
root	16	0.0	0.0	0	0	?	S	17:25	0:00	[migration/0]
root	18	0.0	0.0	0	0	?	S	17:25	0:00	[cpuhp/0]
root	19	0.0	0.0	0	0	?	S	17:25	0:00	[cpuhp/1]
root	20	0.0	0.0	0	0	?	S	17:25	0:00	[migration/1]
root	21	0.0	0.0	0	0	?	S	17:25	0:00	[ksoftirqd/1]
root	23	0.0	0.0	0	0	?	I<	17:25	0:00	[kworker/1:0H-events_highpri]

Cette commande affiche une liste détaillée de tous les processus actifs sur notre système, y compris leur ID de processus (PID), le propriétaire du processus, la consommation de ressources, etc.

Pour fermer un processus spécifique, utiliser la commande "kill" suivie du PID du processus que nous souhaitons arrêter. Par exemple, si vous souhaitez arrêter un processus avec le PID 1234, vous pouvez exécuter la commande :

kill 1234

Si un processus ne se termine pas correctement ou si vous devez le forcer à s'arrêter, vous pouvez utiliser la commande "kill" avec l'option "-9" pour envoyer un signal KILL au processus. Cela force

l'arrêt immédiat du processus sans lui laisser la possibilité de se terminer proprement. Voici comment exécuter cette commande :

kill -9 PID

En comparant la terminaison normale d'un processus avec celle forcée en utilisant le signal KILL (-9), la terminaison normale permet au processus de libérer correctement les ressources et d'exécuter des actions de nettoyage avant de se terminer. En revanche, la terminaison forcée avec le signal KILL ne laisse pas au processus l'opportunité de se terminer proprement, ce qui peut entraîner une perte de données ou des incohérences dans certains cas.

« Il est recommandé d'utiliser d'abord la terminaison normale avant d'avoir recours à la terminaison forcée. »

Job 06 Surveillance des ressources système

Pour surveiller en temps réel l'utilisation du CPU, de la mémoire et d'autres ressources système dans le terminal, utiliser des commandes telles que "top" .

Ces outils affichent des informations détaillées sur les processus en cours d'exécution, l'utilisation des ressources et d'autres statistiques système en temps réel.

```
dome@debianLaPlateformeSSS:~$ top
mon_texte.txt
top - 18:11:12 up 45 min,  1 user,  load average: 0,16, 0,13, 0,09
Tâches: 169 total,  1 en cours, 168 en veille,  0 arrêté,  0 zombie
%Cpu(s):  0,3 ut,  0,2 sy,  0,0 ni, 99,5 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :  1932,4 total,   819,4 libr,   742,0 util,   531,5 tamp/cache
MiB Éch :   975,0 total,   975,0 libr,    0,0 util.  1190,4 dispo Mem
```

PID	UTIL.	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPS+	COM.
370	root	20	0	167720	11476	7968	S	0,3	0,6	0:04.00	vmtoolsd
806	root	20	0	456728	128064	65832	S	0,3	6,5	0:11.91	Xorg
1209	dome	20	0	140100	38552	30368	S	0,3	1,9	0:04.15	vmtoolsd
2361	root	20	0	0	0	0	D	0,3	0,0	0:00.85	kworker/1:0+usb_hub_wq
2363	dome	20	0	11828	5592	3428	R	0,3	0,3	0:00.08	top
1	root	20	0	167764	12296	9224	S	0,0	0,6	0:00.84	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.01	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns

Pour enregistrer ces informations dans un fichier CSV, rediriger la sortie de la commande vers un fichier en utilisant la commande ">" pour écrire et ">>" pour ajouter à un fichier existant.

Voici un exemple pour enregistrer les informations de "top" dans un fichier CSV :

top -b -n 1 | tail -n +8 > ressources_systeme.csv

Cette commande exécute "top" en mode batch (-b) pour une sortie adaptée à un fichier, avec une seule itération (-n 1) et exclut les premières lignes inutiles avec "tail -n +8". Ensuite, elle redirige la sortie vers un fichier CSV nommé "ressources_systeme.csv".

```
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ top -b -n 1 | tail -n +8 > ressources_systeme.csv
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$
```

Une fois que vous avez enregistré ces informations dans un fichier CSV, vous pouvez les ouvrir et les analyser à l'aide d'un programme de feuille de calcul ou d'un script Python pour une analyse plus approfondie.

Job 07 Scripting avancé

Ici un exemple de script Shell pour automatiser la sauvegarde périodique du répertoire "Plateforme" et gérer l'historique des sauvegardes.

```
#!/bin/bash

# Répertoire à sauvegarder
repertoire_source="./Plateforme"

# Répertoire de sauvegarde
repertoire_sauvegarde="./Plateforme/historique"

# Création d'un répertoire pour l'historique des sauvegardes s'il n'existe pas
if [ ! -d "$repertoire_sauvegarde/historique" ]; then
    mkdir -p "$repertoire_sauvegarde/historique"
fi

# Nom du fichier de sauvegarde avec horodatage
nom_fichier_sauvegarde="sauvegarde_$(date +%Y-%m-%d_%H-%M-%S').tar.gz"

# Archivage du répertoire source
tar -zcvf "$repertoire_sauvegarde/$nom_fichier_sauvegarde" "$repertoire_source"

# Copie de la sauvegarde dans le répertoire d'historique
mv "$repertoire_sauvegarde/$nom_fichier_sauvegarde" "$repertoire_sauvegarde/historique/"

echo "Sauvegarde effectuée avec succès."
```

Avant d'exécuter penser à rendre exécutable le script avec la commande suivante:

chmod +x sauvegarde.sh

```
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ chmod +x sauvegarde.sh
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ ls -l
total 28
-rw-r--r-- 1 dome dome   87 13 mars 17:30 personnes.csv
-rw-r--r-- 1 root root  281 13 mars 17:15 personnes.py
-rw-r--r-- 1 dome dome 13525 13 mars 18:15 ressources_systeme.csv
-rwxr-xr-x 1 dome dome   778 13 mars 18:40 sauvegarde.sh
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ ./sauvegarde.sh
```


Pour finir executer le script avec la commande suivante :

./ficher.sh

Et voici le résultat obtenu

```
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ ./sauvegarde.sh
./Plateforme/
./Plateforme/mon_texte_4.txt
./Plateforme/mon_texte_5.txt
./Plateforme/mon_texte.txt
./Plateforme/mon_texte_2.txt
./Plateforme/historique/
./Plateforme/historique/historique/
./Plateforme/mon_texte_3.txt
Sauvegarde effectuée avec succès.
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ ls
personnes.csv  personnes.py  Plateforme  ressources_systeme.csv  sauvegarde.sh
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ cd Plateforme/
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité/Plateforme$ LS
bash: LS : commande introuvable
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité/Plateforme$ ls
historique  mon_texte_2.txt  mon_texte_3.txt  mon_texte_4.txt  mon_texte_5.txt  mon_texte.txt
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité/Plateforme$ cd historique/
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité/Plateforme/historique$ ls
historique
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité/Plateforme/historique$ cd historique/
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité/Plateforme/historique/historique$ ls
sauvegarde_2024-03-13_19-34-05.tar.gz
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité/Plateforme/historique/historique$
```

Job 08 Automatisation des mises à jour logicielles

Voici un exemple de script Shell automatisant la recherche de mises à jour des logiciels existants sur le système :

```
dome@debianLaPlateformeSSS: ~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité
Fichier Édition Affichage Recherche Terminal Aide
GNU nano 7.2 system-update.sh *
#!/bin/bash

# Mettre à jour la liste des paquets disponibles
sudo apt update

# Afficher les paquets pouvant être mis à jour
echo "Paquets pouvant être mis à jour :"
apt list --upgradable

# Demander à l'utilisateur s'il souhaite mettre à jour les paquets
read -p "Voulez-vous mettre à jour les paquets ? (o/n) " choix

if [ "$choix" = "o" ]; then
    # Mettre à jour les paquets
    sudo apt upgrade
    echo "Mise à jour des paquets effectuée avec succès."
else
    echo "Aucune mise à jour n'a été effectuée."
fi
```

Comme dans l'exercice précédent, avant d'exécuter penser à rendre exécutable le script avec la commande suivante :

chmod +x syteme-update.sh

```
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ ls -l
total 36
-rw-r--r-- 1 dome dome 87 13 mars 17:30 personnes.csv
-rw-r--r-- 1 root root 281 13 mars 17:15 personnes.py
drwxr-xr-x 3 dome dome 4096 13 mars 19:34 Plateforme
-rw-r--r-- 1 dome dome 13525 13 mars 18:15 ressources_systeme.csv
-rwxr-xr-x 1 dome dome 767 13 mars 19:33 sauvegarde.sh
-rw-r--r-- 1 dome dome 533 13 mars 21:28 system-update.sh
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ sudo chmod +x system-update.sh
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ ls -l
total 36
-rw-r--r-- 1 dome dome 87 13 mars 17:30 personnes.csv
-rw-r--r-- 1 root root 281 13 mars 17:15 personnes.py
drwxr-xr-x 3 dome dome 4096 13 mars 19:34 Plateforme
-rw-r--r-- 1 dome dome 13525 13 mars 18:15 ressources_systeme.csv
-rwxr-xr-x 1 dome dome 767 13 mars 19:33 sauvegarde.sh
-rwxr-xr-x 1 dome dome 533 13 mars 21:28 system-update.sh
```

Puis exécuter :

./ syteme-update.sh

```
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ ./system-update.sh
Ign :1 cdrom://[Debian GNU/Linux 12.4.0 _Bookworm_ - Official amd64 DVD Binary-1 with firmware 20231210-17:57] bookworm InRelease
Err :2 cdrom://[Debian GNU/Linux 12.4.0 _Bookworm_ - Official amd64 DVD Binary-1 with firmware 20231210-17:57] bookworm Release
Veuillez utiliser apt-cdrom afin de faire reconnaître ce cdrom par votre APT. apt-get update ne peut être employé pour ajouter de nouveaux cdroms
Atteint :3 http://deb.debian.org/debian bookworm InRelease
Réception de :4 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Réception de :5 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Réception de :6 http://security.debian.org/debian-security bookworm-security/main Sources [83,8 kB]
Réception de :7 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [145 kB]
Lecture des listes de paquets... Fait
E: Le dépôt cdrom://[Debian GNU/Linux 12.4.0 _Bookworm_ - Official amd64 DVD Binary-1 with firmware 20231210-17:57] bookworm Release n'a pas de fichier Release.
N: Les mises à jour depuis un tel dépôt ne peuvent s'effectuer de manière sécurisée, et sont donc désactivées par défaut.
N: Voir les pages de manuel d'apt-secure(8) pour la création des dépôts et les détails de configuration d'un utilisateur.
Paquets pouvant être mis à jour :
En train de lister... Fait
Voulez-vous mettre à jour les paquets ? (o/n) o
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
Mise à jour des paquets effectuée avec succès.
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$
```

Job 09 Gestion des dépendances logicielles

Pour simplifier l'installation et la gestion des dépendances logicielles pour un projet web, tout en assurant la compatibilité entre les différentes versions, créer un script Shell qui installe les éléments requis.

Voici un exemple de script qui installe les composants mentionnés :

```
#!/bin/bash

# Installation d'un serveur Web (Apache ou Nginx)
echo "Installation du serveur Web..."
sudo apt update
sudo apt install -y apache2      # Pour Apache
# sudo apt install -y nginx      # Pour Nginx

# Installation d'un système de gestion de base de données relationnelle (MySQL ou MariaDB)
echo "Installation de MariaDB..."
sudo apt install -y mariadb-server

# Installation de phpMyAdmin
echo "Installation de phpMyAdmin..."
sudo apt install -y phpmyadmin

# Installation de Node.js et npm
echo "Installation de Node.js et npm..."
curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
sudo apt install -y nodejs

# Installation de Git
echo "Installation de Git..."
sudo apt install -y git

echo "Installation terminée."
```

Ce script modifié installe d'abord Nginx, puis MariaDB, phpMyAdmin, Node.js avec npm et Git.

Job 10 Sécuriser ses scripts

Procédez à la sécurisation des différents scripts développés précédemment :

Script n.1 system-update.sh:

```
GNU nano 7.2                                system-update.sh
#!/bin/bash

# Mettre à jour la liste des paquets disponibles
sudo apt update

# Afficher les paquets pouvant être mis à jour
echo "Paquets pouvant être mis à jour :"
apt list --upgradable

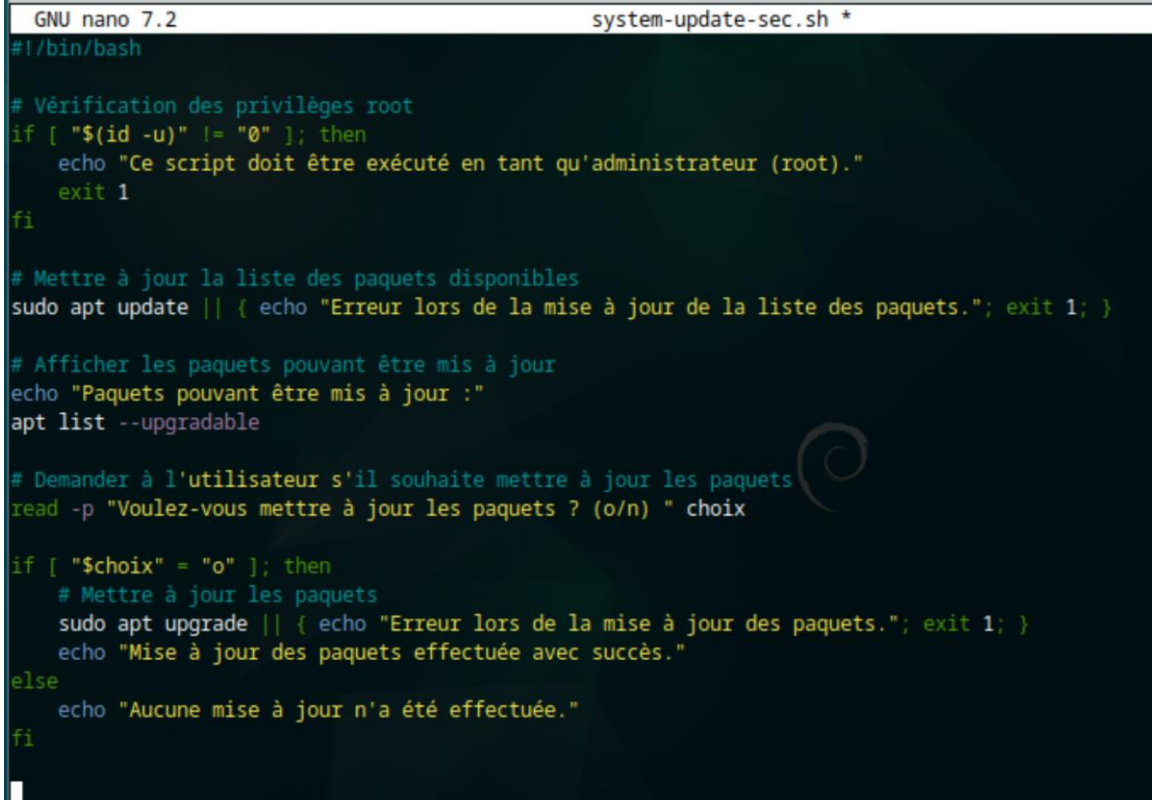
# Demander à l'utilisateur s'il souhaite mettre à jour les paquets
read -p "Voulez-vous mettre à jour les paquets ? (o/n) " choix

if [ "$choix" = "o" ]; then
    # Mettre à jour les paquets
    sudo apt upgrade
    echo "Mise à jour des paquets effectuée avec succès."
else
    echo "Aucune mise à jour n'a été effectuée."
fi
```

ici une version plus sécurisée « system-update-sec.sh » :

```
GNU nano 7.2                                system-update-sec.sh *
#!/bin/bash

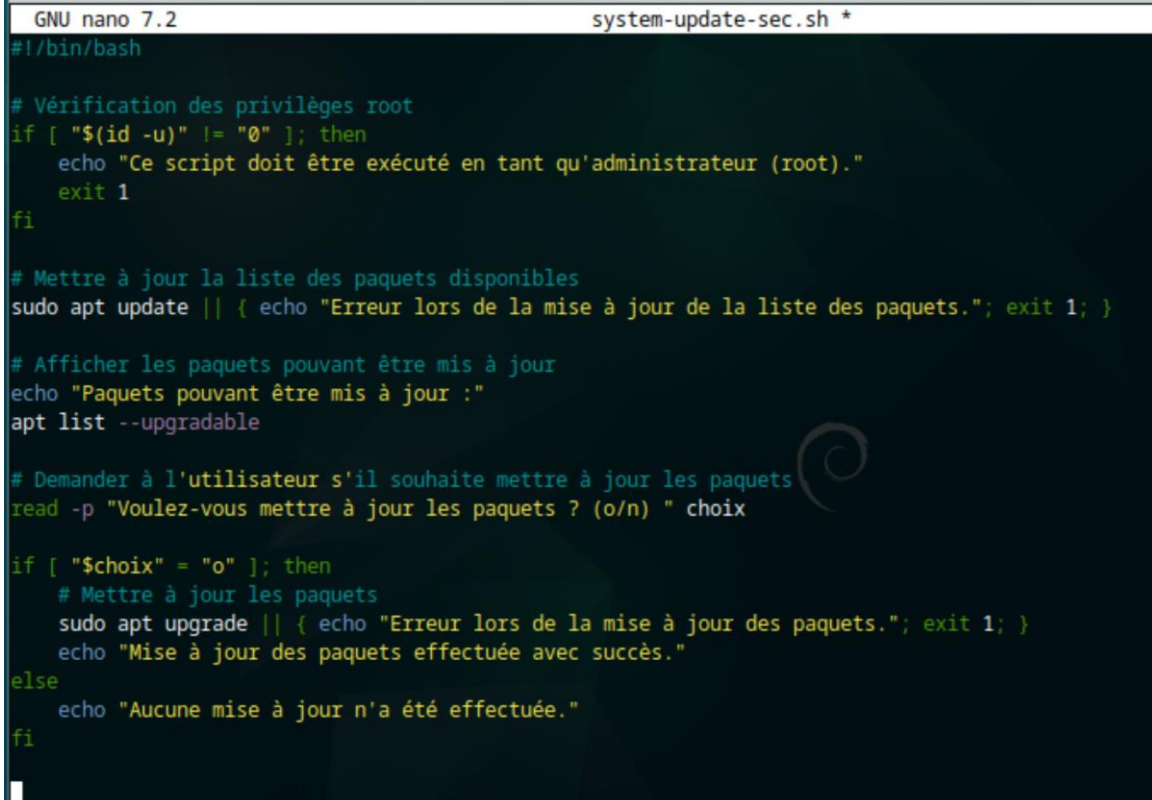
# Vérification des privilèges root
if [ "$(id -u)" != "0" ]; then
    echo "Ce script doit être exécuté en tant qu'administrateur (root)."
```



```
    exit 1
fi

# Mettre à jour la liste des paquets disponibles
sudo apt update || { echo "Erreur lors de la mise à jour de la liste des paquets."; exit 1; }

# Afficher les paquets pouvant être mis à jour
echo "Paquets pouvant être mis à jour :"
```



```
apt list --upgradable

# Demander à l'utilisateur s'il souhaite mettre à jour les paquets
read -p "Voulez-vous mettre à jour les paquets ? (o/n) " choix

if [ "$choix" = "o" ]; then
    # Mettre à jour les paquets
    sudo apt upgrade || { echo "Erreur lors de la mise à jour des paquets."; exit 1; }
    echo "Mise à jour des paquets effectuée avec succès."
else
    echo "Aucune mise à jour n'a été effectuée."
fi
```

Dans la version améliorée du script de mise à jour système, voici comment chaque point a été abordé pour renforcer la sécurité du script :

Validation des privilèges root :

- La vérification des privilèges root a été ajoutée au début du script en utilisant la commande `id -u` pour s'assurer que le script est exécuté en tant qu'administrateur (root). Cela garantit que seules les autorisations nécessaires sont accordées pour exécuter les commandes système.

Gestion des erreurs :

- Des mécanismes de gestion des erreurs ont été intégrés après chaque commande importante (`apt update` et `apt upgrade`). En cas d'échec lors de la mise à jour de la liste des paquets ou de la mise à jour des paquets, le script affiche un message d'erreur explicite et arrête l'exécution avec `exit 1`.

Sécurisation de l'entrée utilisateur :

- L'entrée utilisateur est sécurisée en utilisant la commande `read -p` pour demander à l'utilisateur s'il souhaite mettre à jour les paquets. La réponse est stockée dans la variable `choix` et vérifiée avant de procéder à la mise à jour.


```
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ sudo ./system-update-sec.sh
Atteint :1 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
Paquets pouvant être mis à jour :
En train de lister... Fait
Voulez-vous mettre à jour les paquets ? (o/n) o
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
Mise à jour des paquets effectuée avec succès.
```

Script n .2 sauvegarde-sec.sh :

```
#!/bin/bash

# Répertoire à sauvegarder
repertoire_source="./Plateforme"

# Répertoire de sauvegarde
repertoire_sauvegarde="./Plateforme/historique"

# Création d'un répertoire pour l'historique des sauvegardes s'il n'existe pas
if [ ! -d "$repertoire_sauvegarde/historique" ]; then
    mkdir -p "$repertoire_sauvegarde/historique"
fi

# Nom du fichier de sauvegarde avec horodatage
nom_fichier_sauvegarde="sauvegarde_$(date +%Y-%m-%d_%H-%M-%S').tar.gz"

# Archivage du répertoire source
tar -zcvf "$repertoire_sauvegarde/$nom_fichier_sauvegarde" "$repertoire_source"

# Copie de la sauvegarde dans le répertoire d'historique
mv "$repertoire_sauvegarde/$nom_fichier_sauvegarde" "$repertoire_sauvegarde/historique/"

echo "Sauvegarde effectuée avec succès."
```

ici une version plus sécurisé « sauvegarde-sec.sh »:

```
GNU nano 7.2 sauvegarde-sec.sh *
#!/bin/bash personnel
de dome
# Répertoire à sauvegarder
repertoire_source="./Plateforme"

# Répertoire de sauvegarde
repertoire_sauvegarde="./Plateforme/historique"
mon_texte.txt
# Vérification et création des répertoires s'ils n'existent pas
if [ ! -d "$repertoire_source" ] || [ ! -d "$repertoire_sauvegarde" ]; then
    echo "Erreur : Répertoire source ou répertoire de sauvegarde manquant."
    exit 1
fi

# Création d'un répertoire pour l'historique des sauvegardes s'il n'existe pas
if [ ! -d "$repertoire_sauvegarde/historique" ]; then
    mkdir -p "$repertoire_sauvegarde/historique"
fi

# Nom du fichier de sauvegarde avec horodatage sécurisé
nom_fichier_sauvegarde="sauvegarde_$(date +%Y-%m-%d_%H-%M-%S').tar.gz"


# Archivage du répertoire source avec vérification
tar -zcvf "$repertoire_sauvegarde/$nom_fichier_sauvegarde" "$repertoire_source" || { echo "Erreur lors de l'archivage."; exit 1; }

# Copie sécurisée de la sauvegarde dans le répertoire d'historique
mv "$repertoire_sauvegarde/$nom_fichier_sauvegarde" "$repertoire_sauvegarde/historique/" || { echo "Erreur lors de la copie dans l'historique."; exit 1; }

echo "Sauvegarde effectuée avec succès."
```

```
GNU nano 7.2                                web-project-sec.sh
#!/bin/bash

# Vérification des privilèges root
if [ "$(id -u)" != "0" ]; then
    echo "Ce script doit être exécuté en tant qu'administrateur (root)."
```



```
    exit 1
fi

# Installation d'un serveur Web Nginx
echo "Installation de Nginx..."
sudo apt update
sudo apt install -y nginx

# Installation d'un système de gestion de base de données relationnelle MariaDB
echo "Installation de MariaDB..."
sudo apt install -y mariadb-server

# Installation de phpMyAdmin
echo "Installation de phpMyAdmin..."
sudo apt install -y phpmyadmin

# Installation de Node.js et npm
echo "Installation de Node.js et npm..."
curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
sudo apt install -y nodejs

# Installation de Git
echo "Installation de Git..."
sudo apt install -y git

echo "Installation terminée en toute sécurité."
```

Dans le script de sauvegarde amélioré, plusieurs mesures de sécurité et de robustesse ont été mises en place :

Vérification des répertoires source et de sauvegarde :

- Avant de commencer la sauvegarde, le script vérifie si les répertoires source et de sauvegarde existent. Cela permet d'éviter les erreurs en cas d'absence de ces répertoires.

Création du répertoire d'historique :

- Le script crée un répertoire pour stocker les sauvegardes historiques s'il n'existe pas. Cela garantit que les sauvegardes sont organisées et faciles à gérer.

Gestion des erreurs avec messages explicites :

- Des mécanismes de gestion des erreurs ont été ajoutés pour traiter les cas où la création de répertoires ou l'archivage échouent. Des messages d'erreur explicites sont affichés pour informer l'utilisateur en cas de problème.

```
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ ./sauvegarde-sec.sh
./Plateforme/
./Plateforme/mon_texte_4.txt
./Plateforme/mon_texte_5.txt
./Plateforme/mon_texte.txt
./Plateforme/mon_texte_2.txt
./Plateforme/historique/
./Plateforme/historique/sauvegarde_2024-03-15_15-51-30.tar.gz
./Plateforme/historique/historique/
./Plateforme/historique/historique/sauvegarde_2024-03-13_19-34-05.tar.gz
./Plateforme/mon_texte_3.txt
Sauvegarde effectuée avec succès.
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$
```

Script n . 3 « web-project.sh »:

```
GNU nano 7.2 web-project.sh
#!/bin/bash

# Installation d'un serveur Web (Apache ou Nginx)
echo "Installation du serveur Web..."
sudo apt update
sudo apt install -y apache2 # Pour Apache
# sudo apt install -y nginx # Pour Nginx

# Installation d'un système de gestion de base de données relationnelle (MySQL ou MariaDB)
echo "Installation de MariaDB..."
sudo apt install -y mariadb-server

# Installation de phpMyAdmin
echo "Installation de phpMyAdmin..."
sudo apt install -y phpmyadmin

# Installation de Node.js et npm
echo "Installation de Node.js et npm..."
curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
sudo apt install -y nodejs

# Installation de Git
echo "Installation de Git..."
sudo apt install -y git

echo "Installation terminée."
```

ici une version plus sécurisé « web-project-sec.sh »

```
GNU nano 7.2                                web-project.sh
#!/bin/bash

# Installation d'un serveur Web (Apache ou Nginx)
echo "Installation du serveur Web..."
sudo apt update
sudo apt install -y apache2    # Pour Apache
# sudo apt install -y nginx    # Pour Nginx

# Installation d'un système de gestion de base de données relationnelle (MySQL ou MariaDB)
echo "Installation de MariaDB..."
sudo apt install -y mariadb-server

# Installation de phpMyAdmin
echo "Installation de phpMyAdmin..."
sudo apt install -y phpmyadmin

# Installation de Node.js et npm
echo "Installation de Node.js et npm..."
curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
sudo apt install -y nodejs

# Installation de Git
echo "Installation de Git..."
sudo apt install -y git

echo "Installation terminée."
```

Pour sécuriser le script d'installation des composants (Nginx, MariaDB, phpMyAdmin, Node.js, Git), voici comment chaque point a été abordé :

- **Vérification des privilèges root :**

La première étape du script vérifie si l'utilisateur exécutant le script a les privilèges root en utilisant la commande `id -u`. Cela garantit que seules les autorisations nécessaires sont accordées pour l'installation des composants.

- **Utilisation de sudo de manière ciblée :**

Les commandes nécessitant des privilèges élevés, telles que `apt update`, `apt install`, et l'utilisation de `curl` pour Node.js, sont précédées de `sudo`. Cela permet de limiter l'exécution de ces commandes à des utilisateurs autorisés.

- **Installation sécurisée des composants :**

Chaque étape d'installation est clairement indiquée avec un message informatif pour suivre le processus. Cela permet à l'utilisateur de savoir quel composant est en cours d'installation.

- **Finalisation sécurisée :**

Le script se termine par un message indiquant que l'installation s'est déroulée en toute sécurité, offrant une confirmation visuelle à l'utilisateur que toutes les étapes ont été exécutées correctement.

```
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ ./web-project-sec.sh
Ce script doit être exécuté en tant qu'administrateur (root).
dome@debianLaPlateformeSSS:~/Documents/Projets_La_Plateforme/Système-Scripts-Sécurité$ sudo ./web-project-sec.sh
Installation de Nginx...
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Atteint :2 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
Lecture des listes de paquets... Fait
```

Job 11 Utilisation d'API Web dans un script :