

## **REVISION SYSTÈME**

GACI Hanafi  
DERRI Tara  
BOTTERO Mathis

### **ETAPE 1 : CRÉATION DE LA MACHINE VIRTUELLE :**

Pour notre projet, commençons par choisir un logiciel de virtualisation adapté à nos besoins. Nous avons plusieurs options telles que VirtualBox, VMware ou Parallels Desktop. Chacune de ces solutions offre des fonctionnalités similaires, ici nous avons choisi VMware

Maintenant que nous avons sélectionné notre logiciel, il est temps de télécharger l'image d'installation de Debian. Selon l'architecture de notre processeur, nous avons choisi l'image AMD64 .

Maintenant que notre machine virtuelle est configurée, attachons l'image d'installation de Debian et démarrons-la. Suivons les instructions d'installation de Debian à l'écran. Pendant le processus d'installation, nous serons invités à créer un utilisateur avec un mot de passe. qui sera "La\_Platforme" et le mot de passe "LAPlateforme\_".

### **ETAPE 2 : Commandes de recherche avancée**

Pour notre prochaine tâche, nous devons créer cinq fichiers texte contenant le texte « Que la force soit avec toi. » et les répartir dans différents répertoires de notre système. Ensuite, nous devrons utiliser le terminal pour localiser ces fichiers en utilisant le mot-clé "force".

Commençons par créer les fichiers et les placer dans les répertoires spécifiés. Voici les commandes que nous devons utiliser :

```
echo "Que la force soit avec toi." > Bureau/mon_texte.txt
```

```
echo "Que la force soit avec toi." > Documents/mon_texte.txt
```

```
echo "Que la force soit avec toi." > Téléchargement/mon_texte.txt
```

```
echo "Que la force soit avec toi." > Vidéos/mon_texte.txt
```

```
echo "Que la force soit avec toi." > Images/mon_texte.txt
```

Ensuite, nous devons utiliser la commande grep pour rechercher les fichiers contenant le mot "force". Voici comment nous allons le faire :

```
grep -l "force" Bureau/mon_texte.txt Documents/mon_texte.txt  
Téléchargement/mon_texte.txt Vidéos/mon_texte.txt Images/mon_texte.txt
```

### ETAPE 3 : Compression et décompression de fichiers

Maintenant, créons le répertoire "Plateforme" dans le dossier "Documents" de notre session et ajoutons-y le fichier "mon\_texte.txt". Ensuite, nous allons dupliquer ce fichier quatre fois dans le même répertoire, formant ainsi un total de cinq fichiers dans le répertoire "Plateforme".

Voici comment nous pouvons faire cela en utilisant le terminal:

```
mkdir Documents/Plateforme
```

```
cp Documents/mon_texte.txt Documents/Plateforme/
```

```
cd Documents/Plateforme
```

Maintenant que nous avons notre répertoire "Plateforme" avec cinq fichiers, passons à l'étape suivante : archiver le répertoire en utilisant les commandes suivantes :

```
tar -czvf Plateforme.tar.gz *
```

Cette commande va créer une archive nommée "Plateforme.tar.gz" contenant tous les fichiers du répertoire "Plateforme" et les compresser en utilisant gzip.

Maintenant, pour décompresser les archives, nous utilisons les commandes suivantes :

```
tar -xzvf Plateforme.tar.gz
```

```
tar -xzvf Plateforme.tar.gz
```

Cette commande va extraire le contenu de l'archive dans le répertoire actuel.

## **ETAPE 4 : Manipulation de texte**

nous allons commencer par créer un script Python pour générer un fichier CSV avec les données fournies

```
with open('personnes.csv', 'w') as f:  
    f.write('Nom, Âge, Ville\n')  
  
    f.write('Jean, 25 ans, Paris\n')  
  
    f.write('Marie, 30 ans, Lyon\n')  
  
    f.write('Pierre, 22 ans, Marseille\n')  
  
    f.write('Sophie, 35 ans, Toulouse\n')
```

Ce script crée un fichier "personnes.csv" et y ajoute les données fournies.

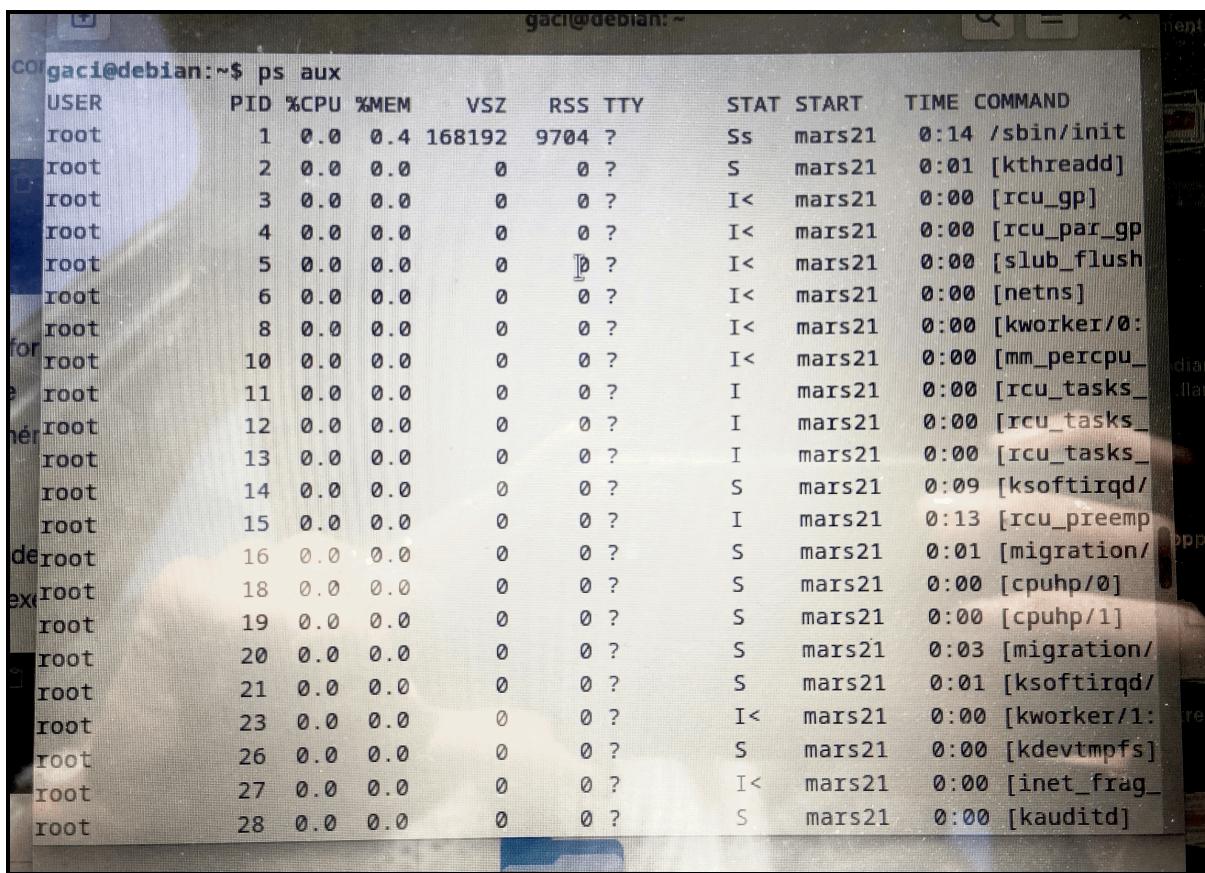
Maintenant, utilisons la commande "awk" pour extraire les informations relatives aux villes de chaque personne à partir du fichier CSV :

```
awk -F ',', ''NR > 1 {print $3}' personnes.csv
```

- `-F ',', ''`: spécifie le délimiteur de champ comme étant la virgule suivie d'un espace.
- `NR > 1`: indique à Awk de commencer à traiter à partir de la deuxième ligne, en sautant l'en-tête.
- `{print $3}`: affiche le troisième champ de chaque ligne, qui correspond à la ville.

## ETAPE 5 :Gestion des processus

Pour explorer les processus actifs nous utilisons la commande ps aux



```
gaci@debian:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root      1  0.0  0.4 168192  9704 ?        Ss   mars21  0:14 /sbin/init
root      2  0.0  0.0     0     0 ?        S     mars21  0:01 [kthreadd]
root      3  0.0  0.0     0     0 ?        I<   mars21  0:00 [rcu_gp]
root      4  0.0  0.0     0     0 ?        I<   mars21  0:00 [rcu_par_gp]
root      5  0.0  0.0     0     0 ?        I<   mars21  0:00 [slub_flush]
root      6  0.0  0.0     0     0 ?        I<   mars21  0:00 [netns]
root      8  0.0  0.0     0     0 ?        I<   mars21  0:00 [kworker/0:
for      10 0.0  0.0    0     0 ?       I<   mars21  0:00 [mm_percpu_
e       11 0.0  0.0    0     0 ?       I     mars21  0:00 [rcu_tasks_...
nér      12 0.0  0.0    0     0 ?       I     mars21  0:00 [rcu_tasks_...
root     13 0.0  0.0    0     0 ?       I     mars21  0:00 [rcu_tasks_...
root     14 0.0  0.0    0     0 ?       S     mars21  0:09 [ksoftirqd/...
root     15 0.0  0.0    0     0 ?       I     mars21  0:13 [rcu_preemp...
deroot   16 0.0  0.0    0     0 ?       S     mars21  0:01 [migration/...
exroot   18 0.0  0.0    0     0 ?       S     mars21  0:00 [cpuhp/0]
root     19 0.0  0.0    0     0 ?       S     mars21  0:00 [cpuhp/1]
root     20 0.0  0.0    0     0 ?       S     mars21  0:03 [migration/...
root     21 0.0  0.0    0     0 ?       S     mars21  0:01 [ksoftirqd/...
root     23 0.0  0.0    0     0 ?       I<   mars21  0:00 [kworker/1:...
root     26 0.0  0.0    0     0 ?       S     mars21  0:00 [kdevtmpfs]
root     27 0.0  0.0    0     0 ?       I<   mars21  0:00 [inet_frag_...
root     28 0.0  0.0    0     0 ?       S     mars21  0:00 [kaudit...

```

pour désactiver le processus : Kill suivi du PID voulu

et pour désactiver de force . Kill -9 suivi du PID

la commande kill peut entraîner une perte de données ou d'autres problèmes, car le processus n'a pas la possibilité de se terminer correctement.

## **ETAPE 6 : surveillance des ressources systèmes**

Afficher les informations du CPU: avec la commande top

```
top - 13:13:26 up 20:41, 2 users, load average: 0,53, 1,78, 1,51
Tâches: 342 total, 1 en cours, 341 en veille, 0 arrêté, 0 zombie
%Cpu(s): 10,3 ut, 16,2 sy, 0,0 ni, 72,3 id, 0,0 wa, 0,0 hi, 1,2 si, 0,0 st
MiB Mem : 1901,7 total, 77,9 libr, 1481,2 util, 513,4 tamp/cache
MiB Éch : 977,0 total, 503,5 libr, 473,5 util, 420,5 dispo Mem

 PID UTIL. PR NI VIRT RES SHR S %CPU %MEM TEMPS+ COM.
16116 root 20 0 416112 107848 56576 S 9,5 5,5 2:54.90 packag+
18255 gaci 20 0 11836 5652 3468 R 2,5 0,3 0:00.67 top
14 root 20 0 0 0 0 S 1,3 0,0 0:14.42 ksoftti+
16686 gaci 20 0 3796296 227320 120192 S 1,3 11,7 0:40.86 gnome-+
16931 gaci 20 0 287292 36492 29316 S 0,9 1,9 0:06.09 vmtool+
656 message+ 20 0 18368 11316 4092 S 0,6 0,6 0:26.25 dbus-d+
9420 gacii 20 0 3755936 127100 39984 S 0,6 6,5 4:16.55 gnome-+
17658 gaci 20 0 558264 49956 37696 S 0,6 2,6 0:02.92 gnome-+
18144 root 20 0 0 0 0 I 0,6 0,0 0:00.21 kwork+
15 root 20 0 0 0 0 I 0,3 0,0 0:19.60 rcu_pr+
21 root 20 0 0 0 0 S 0,3 0,0 0:02.34 ksoftti+
36 root 39 19 0 0 0 S 0,3 0,0 0:15.23 khugep+
696 root 20 0 241444 8152 5804 S 0,3 0,4 5:23.67 vmtool+
9591 gacii 20 0 312900 9052 7032 S 0,3 0,5 0:00.97 goa-id+
16911 gaci 20 0 805392 84132 43520 S 0,3 4,3 0:23.86 gnome-+
17710 root 20 0 0 0 0 I 0,3 0,0 0:01.12 kwork+
1 root 20 0 168192 9464 6796 S 0,0 0,5 0:17.15 systemd
```

commande pour enregistrer dans un fichier csv:

```
top -b -n 1 | awk '{print $1,"$9","$10}' > ressources_systeme.csv
```

Ajout de la commande dans crontab:

- \* \* bash /home/la\_plateforme/Documents/Python/script.sh

Pour cet exercice nous avons d'abord créer un script car la commande seul ne fonctionnent pas dans crontab, celui-ci permet de récupérer les information du pu dans un fichier csv grâce a la commande top avec les

paramètre -b -n 1

-b permet d'effectuer un cycle puis quitter, -n 1 permet de mettre a jour grâce a un nombre saisi après. Puis en mettant dans le crontab (crontab -e) la direction du script avec le temps devant celle ci s'effectuera automatiquement

### **ETAPE 7: Scripting avancé:**

Nous avons conçu un script Shell dans le but d'automatiser la sauvegarde périodique du répertoire "Plateforme" dans un fichier TAR, tout en intégrant une fonctionnalité de gestion d'historique des sauvegardes. Tout d'abord, nous définissons les chemins nécessaires pour le répertoire à sauvegarder ainsi que le répertoire où les sauvegardes seront stockées. Ensuite, nous récupérons la date et l'heure actuelle pour générer un nom de fichier de sauvegarde unique. Nous créons également le répertoire de sauvegarde s'il n'existe pas déjà. Pour garantir la traçabilité des sauvegardes effectuées, nous vérifions si le fichier de liste d'archives de sauvegarde existe. S'il n'existe pas, nous le créons. Enfin, nous créons une archive TAR contenant le répertoire à sauvegarder en utilisant le fichier de liste d'archives pour suivre les modifications incrémentielles. Cette approche nous permet de maintenir un historique des sauvegardes effectuées, ce qui est essentiel pour le suivi des opérations et la gestion des données.

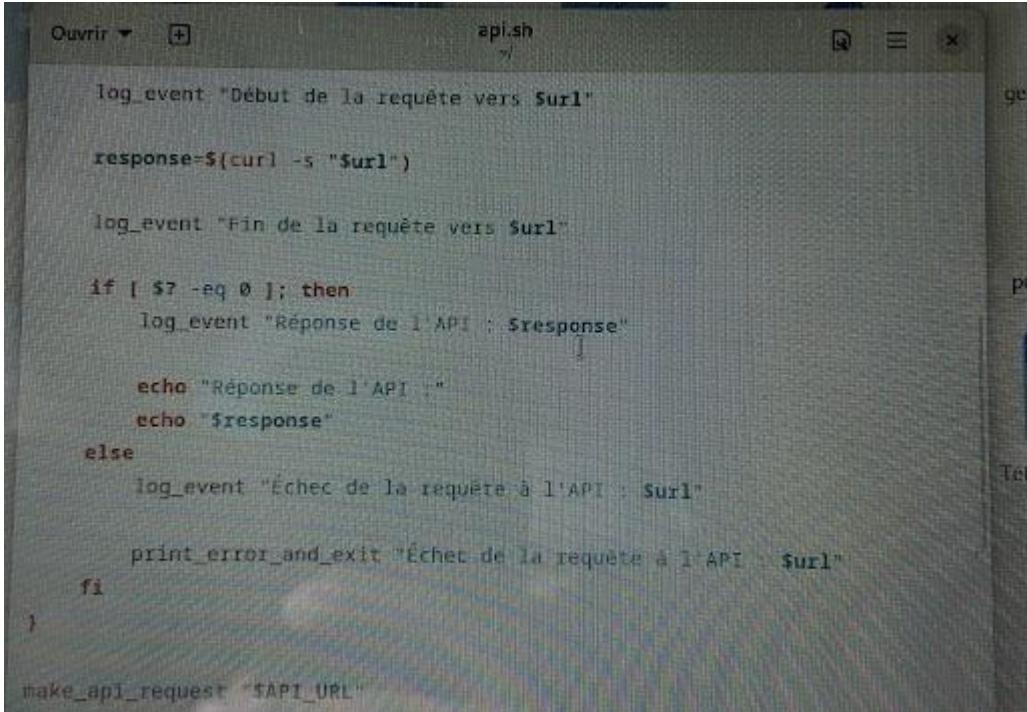
La négligence de la sécurité dans nos scripts peut entraîner divers risques, notamment :

- **Injection de commandes** : Nos scripts pourraient être vulnérables à des attaques par injection de commandes, permettant l'exécution de commandes malveillantes sur le système hôte.
- **Exécution de code arbitraire** : Des acteurs malveillants pourraient exploiter des vulnérabilités dans nos scripts pour exécuter du code arbitraire sur le système hôte, causant ainsi des dommages considérables.
- **Fuites d'informations sensibles** : Nos scripts pourraient contenir des données sensibles telles que des mots de passe ou des clés d'API, dont une gestion insuffisante pourrait entraîner des fuites de données compromettant la sécurité du système.
- **Déni de service** : Des scripts non sécurisés pourraient être utilisés pour des attaques de déni de service, surchargeant les ressources système ou bloquant l'accès à certaines fonctionnalités.

Pour assurer la sécurité des scripts que nous avons développés précédemment, voici quelques bonnes pratiques à adopter :

- **Limiter les priviléges excessifs** : Il est essentiel de restreindre les priviléges des scripts en utilisant des permissions d'exécution appropriées et en évitant d'exécuter des commandes avec des priviléges élevés, comme l'utilisation excessive de sudo.
- **Sécuriser les communications** : Nous devons veiller à utiliser des connexions sécurisées (HTTPS) et à chiffrer les données sensibles lors de leur transmission sur le réseau.
- **Gestion sécurisée des informations sensibles** : Il est recommandé d'éviter de stocker des informations sensibles telles que les mots de passe en clair dans les scripts. L'utilisation de variables d'environnement sécurisées ou de services de gestion des secrets pour stocker ces informations est une pratique recommandée.

## **ETAPE 10:Utilisation d'api web dans un script**



The screenshot shows a terminal window with a dark background and light-colored text. The title bar of the window says "api.sh". The script content is as follows:

```
log_event "Début de la requête vers $url"

response=$(curl -s "$url")

log_event "Fin de la requête vers $url"

if [ $? -eq 0 ]; then
    log_event "Réponse de l'API : $response"
    echo "Réponse de l'API :"
    echo "$response"
else
    log_event "Échec de la requête à l'API : $url"
    print_error_and_exit "Échec de la requête à l'API : $url"
fi

make_api_request "$API_URL"
```

The image shows a terminal window titled "api.sh" with the following content:

```
#!/bin/bash

API_URL="https://api.example.com/data"

log_event() {
    local message="$1"
    local timestamp=$(date +"%Y-%m-%d %T")
    echo "$timestamp - $message" >> api.log
}

print_error_and_exit() {
    local message="$1"
    echo "Erreur : $message" >> 2>
    exit 1
}

make_api_request() {
    local url=$1
    local response=""

    curl -s $url > response
```

