

TP : Saveadmin

Réalisé par : Chams TMAR (GL3-2)

Dans ce TP, on voulait écrire une application shell pour l'administration de la sauvegarde et restauration des fichiers. Ainsi, on va afficher un menu à l'utilisateur, présentant différentes options.

En fonction du choix de l'utilisateur, le script exécute différentes actions à l'aide d'une structure de contrôle `case` :

- Si le choix est **1**, on liste les fichiers les plus volumineux.
- Si le choix est **2**, on fait la sauvegarde ou la restauration avec `tar`.
- Si le choix est **3**, on fait la sauvegarde ou la restauration avec `dd`.
- Si le choix est **4**, on fait la sauvegarde ou la restauration avec `cpio`.
- Si le choix est **5**, on fait la sauvegarde ou la restauration avec `dump/restore`.
- Si le choix est **6**, le script quitte en utilisant la commande `exit`.

Voici le script expliqué :

```
echo "Sauvegarde et Restauration"

while true
do

    echo "1- Lister les fichiers suid"
    echo "2- Lister les fichiers guid"
    echo "3- Lister les répertoires sticky bit"
    echo "4- Lancer un sniffer (tcpdump)"
    echo "5- Gérer un firewall (iptables)"
    echo "6- Quitter"

    echo ""
    echo "Tapez votre choix"
    read choix

    case $choix in
```

1) #on invite l'utilisateur à entrer un répertoire. Si aucun n'est donné, le répertoire actuel (.) est utilisé par défaut.

```
read -p "Donner un dossier (.): " dir
if [[ -z "$dir" ]]
then
    dir=.
fi
```

#on recherche les fichiers (-type f) dans le répertoire spécifié, compte le nombre d'octets pour chaque fichier avec wc -c, puis trie et affiche les 5 fichiers les plus volumineux.

```
find "$dir" -type f -exec wc -c {} \; 2> /dev/null \
| sort -r -n \
| head -5
;;
```

2) echo "1- Save"
echo "2- Restore"

```
read -p "Donner votre choix (1): " choice
#choix par défaut = 1
if [[ -z "$choice" ]]
then
    choice=1
fi
```

```
case $choice in
```

1) #on invite l'utilisateur à entrer le répertoire à sauvegarder (input) et le chemin de sauvegarde (output).

```
read -p "Le dossier à sauvegarder: " input
read -p "Le chemin de sauvegarde: " output
```

#on change de répertoire vers le répertoire d'entrée (cd "\$input") et obtient son chemin absolu.

```
cd "$input"
input=$(pwd)
```

```
cd - > /dev/null
```

```
#on crée une archive tar (tar -cvf "$output" "$input")
```

```

tar -cvf "$output" "$input"
;;
2) read -p "Le chemin de sauvegarde: " input

if [[ ! "$input" = /* ]]
then
    input="$(pwd)/$input"
fi
#si le chemin est relatif, on le rend absolu.
cd /
#on extrait le contenu de l'archive tar.
tar -xvf "$input"

cd - > /dev/null
;;
esac
;;
3) echo "1- Save"
echo "2- Restore"

read -p "Donner votre choix (1): " choice

if [[ -z "$choice" ]]
then
    choice=1
fi

case $choice in
1) #on invite l'utilisateur à entrer le répertoire à
sauvegarder (input) et le chemin de sauvegarde (output).
    read -p "Le dossier à sauvegarder: " input
    read -p "Le chemin de sauvegarde: " output

    cd "$input"
    input=$(pwd)

    cd - > /dev/null

```

```

        #on crée les répertoires de destination
        find "$input" -type d -exec mkdir -p "$output{}" \;
        #on copie les fichiers
        find "$input" -type f -exec dd if="{}"
of="$output{}" \;
        ;;
2)    #on invite l'utilisateur pour le chemin de
sauvegarde
        read -p "Le chemin de sauvegarde: " input

        cd "$input"
        #on crée les répertoires de destination
        find . -type d -exec mkdir -p "/{}" \;
        #on restaure les fichiers
        find . -type f -exec dd if="{}" of="/{}" \;

        cd - > /dev/null
        ;;
esac
;;
4)    echo "1- Save"
        echo "2- Restore"

        read -p "Donner votre choix (1): " choice

        if [[ -z "$choice" ]]
        then
                choice=1
        fi

        case $choice in
1)    read -p "Le dossier à sauvegarder: " input
                read -p "Le chemin de sauvegarde: " output
                #on invite l'utilisateur à entrer le répertoire à
sauvegarder (input) et le chemin de sauvegarde (output).
                cd "$input"
                input=$(pwd)

```

```

cd - > /dev/null
#on crée une archive cpio.
find "$input" | cpio -o > "$output"
;;

2) #on invite l'utilisateur pour le chemin de
sauvegarde (input).
read -p "Le chemin de sauvegarde: " input

if [[ ! "$input" = /* ]]
then
    input="$(pwd)/$input"
fi

cd /
#on extrait l'archive cpio
cpio -iu < "$input"

cd - > /dev/null
;;

esac
;;

5) echo "1- Save"
echo "2- Restore"

read -p "Donner votre choix (1): " choice

if [[ -z "$choice" ]]
then
    choice=1
fi

case $choice in
1) read -p "Le dossier à sauvegarder: " input
    read -p "Le chemin de sauvegarde: " output
    #on utilise dump pour créer une sauvegarde (sudo
dump -0f "$output" "$input").
    sudo dump -0f "$output" "$input"
    ;;

```

```

2)  read -p "Le chemin de sauvegarde: " input

    if [[ ! "$input" = /* ]]
    then
        input="$(pwd)/$input"
    fi

    cd /
    #on utilise restore pour restaurer à partir de la
sauvegarde (sudo restore -rf "$input").
    sudo restore -rf "$input"

    cd - > /dev/null
    ;;
esac
;;
6)  break
;;
esac

done

```