
	<h1 style="text-align: center;">TP-5 : Bonus</h1> <hr style="width: 50%; margin: auto;"/> <p style="text-align: center;">Surveillance de Zone d'Échange</p>	 <p style="text-align: center;">1er année TP5</p>
---	---	--

<p><u>Prérequis :</u></p> <ul style="list-style-type: none"> ▪ Cours Linux, Shell Script, ▪ TP Commande Linux, ▪ TP1 et TP2 Shell Script. ▪ Système : Debian / Raspberry Pi OS. ▪ Accès : Privilèges sudo ou compte root. ▪ Compétences : Manipulation du Shell Bash et commandes de base. 	<p><u>Matériels et documentation :</u></p> <ul style="list-style-type: none"> ▪ DEBIAN ▪ Raspberry PI400 	<p><u>Durée :</u></p> <p style="font-size: 1.5em; text-align: center;">2H00</p>
---	---	---

Objectifs	<p>À la fin de la séance, vous devrez être capable de :</p> <ul style="list-style-type: none"> ▪ Ecrire des scripts d'administration d'utilisateurs UNIX. ▪ Programmer l'exécution automatique de scripts.
------------------	--

1 Scénario

L'entreprise **GigaFret** dispose d'un dossier d'échange /tmp/transfert. Les clients y déposent des fichiers CSV de commandes via FTP. Votre mission est de créer un script "Gardien" (gardien.sh) qui tourne en arrière-plan pour valider ces fichiers, les ranger par utilisateur et supprimer les intrus.

1.1 Préparation :

- Dans votre répertoire, créer le dossier TP5
- Dans TP5, créer le dossier /tmp/transfert
- Créer dans le dossier `transfert` 6 fichiers :
 - Fichier1.csv contenant la date + la température du processeur
 - Fichier2.csv contenant `Technicien;toto` + la date
 - Fichier3.csv vide
 - Fichier4.txt contenant un lorem de 45 caractères sur 3 lignes
 - Fichier5.log contenant le nom de l'utilisateur loggé et la date
 - Fichier6.sys contenant les infos de la mémoire ram

1.2 Les consignes (Le défi)

Le script doit scanner le dossier /tmp/transfert.

- Si un fichier n'est pas un .csv, il doit être immédiatement déplacé dans un dossier /tmp/quarantaine.
- Si un fichier .csv est vide (0 octet), il doit être supprimé.

Ce script doit être structuré comme suit :

```
#!/bin/bash

# *****
# Nom: surveillance_nettoyage.sh
# Description: Scan, filtrage et nettoyage du dossier transfert
# *****
```

1.2.1 Définition des dossiers

```
SOURCE="/tmp/transfert"
QUARANTAINE="/tmp/quarantaine"
```

1.2.2 Création des dossiers s'ils n'existent pas

```
if [ ! -d "$SOURCE" ]; then
  mkdir -p "$SOURCE"
fi

if [ ! -d "$QUARANTAINE" ]; then
  mkdir -p "$QUARANTAINE"
fi

echo "Analyse du dossier $SOURCE en cours..."
```

1.2.3 Boucle de scan (on parcourt tous les fichiers du dossier)

```
for fichier in "$SOURCE"/*
do
# Sécurité : Vérifier si le dossier est
[ -e "$fichier" ] || continue # passe au fichier suivant

# On récupère juste le nom du fichier (sans le chemin /tmp/transfert/)
NOM_FICH=$(basename "$fichier")

# Test de l'extension : Si ce n'est pas un .csv
if [[ ! "$NOM_FICH" == *.csv ]]; then
echo "Alerte : $NOM_FICH n'est pas un CSV. Déplacement en quarantaine."
mv "$fichier" "$QUARANTAINE/"
continue # passe au fichier suivant
fi

# Test de la taille : Si le fichier .csv est vide (0 octet)
# L'option -s vérifie si le fichier a une taille supérieure à 0
if [ ! -s "$fichier" ]; then
echo "Nettoyage : $NOM_FICH est vide. Suppression."
rm "$fichier"
```

```
else
    echo "Validation : $NOM_FICH est conforme (format CSV et non vide)."
```

fi

done

```
echo "Traitement terminé."
```

1.2.4 Traitement Intelligent (Extraction & Création) :

Chaque fichier CSV valide contient le nom d'un technicien dans sa première ligne (ex: Technicien;Nom).

- Le script doit extraire ce nom.
- Si l'utilisateur (ex: *nom*) n'existe pas sur la Raspberry Pi, le script doit **le créer automatiquement** (en utilisant `useradd -m`, comme vu dans le TP4).

1.3 Archivage Sécurisé :

- Le script doit créer un dossier `Commandes` dans le `/home` du technicien concerné.
- Il doit y déplacer le fichier CSV et changer les permissions pour que **seul ce technicien** puisse le lire (`chmod 600`).

2 Annexe :

Voici quelques scripts utiles

- **Vérifier si un utilisateur existe :**

```
if id "$NOM_TECH" &>/dev/null; then ...
```

- **Extraire le nom du technicien (Ligne 1, colonne 2) :**

```
NOM_TECH=$(head -n 1 "$fichier" | cut -d';' -f2 | tr '[:upper:]' '[:lower:]')
```

- **Vérifier la taille d'un fichier :**

```
if [ ! -s "$fichier" ]; then rm "$fichier"; fi
```

- **Générateur de fichiers :**

```
#!/bin/bash

DEST="/tmp/transfert"
mkdir -p "$DEST"

echo "Génération des fichiers de test dans $DEST..."

/sys/class/thermal/thermal_zone0/temp
TEMP=$(cat /sys/class/thermal/thermal_zone0/temp)
TEMP_C=$((TEMP / 1000))
echo "$(date +%Y-%m-%d);${TEMP_C}°C" > "$DEST/Fichier1.csv"

echo "Technicien;toto" > "$DEST/Fichier2.csv"
echo "$(date +%Y-%m-%d)" >> "$DEST/Fichier2.csv"

touch "$DEST/Fichier3.csv"

echo "Lorem ipsum dolor sit amet, consectetur elit." > "$DEST/Fichier4.txt"
echo "Sed do eiusmod tempor incididunt ut labore p." >> "$DEST/Fichier4.txt"
echo "Ut enim ad minim veniam, quis nostrud exerc." >> "$DEST/Fichier4.txt"

echo "Utilisateur: $USER" > "$DEST/Fichier5.log"
echo "Date: $(date)" >> "$DEST/Fichier5.log"

free -h > "$DEST/Fichier6.sys"

echo "Tous les fichiers ont été générés avec succès."
ls -l "$DEST"
```