Externaliser les sauvegardes (SSH & Rsync)

Introduction : Le scénario "Incendie"

Tu as tes sauvegardes sur le serveur GLPI. C'est bien.

Mais imagine que la salle serveur prenne feu, ou que le disque dur physique du serveur lâche. Tu as perdu ton serveur GLPI ET tes sauvegardes.

L'objectif du jour :

Faire en sorte que ton serveur GLPI envoie automatiquement, chaque nuit, une copie de ses sauvegardes vers un autre serveur (un serveur de stockage ou de backup) situé ailleurs.

Pour cela, nous allons utiliser deux outils légendaires de Linux :

1. **SSH (Secure Shell) :** Pour créer un tunnel sécurisé entre les deux serveurs.
2. **Rsync :** Pour synchroniser les fichiers intelligemment.

Prérequis

- **Serveur A (Source)** : Ton serveur GLPI (celui qui a les données).
- **Serveur B (Destination)** : Une autre machine Linux (une autre VM) qui servira de coffre-fort. Note son adresse IP (ex: 192.168.1.50).

Partie 1 : L'authentification sans mot de passe

C'est le défi principal.

Si tu lances une copie manuelle, le serveur B va te demander un mot de passe.

Mais notre script tourne la nuit, tout seul. Il ne peut pas taper de mot de passe !

Nous allons utiliser le principe des **Clés SSH**.

L'analogie de la clé et du cadenas :

- Nous allons fabriquer une clé unique (Clé Privée) que nous gardons sur le serveur GLPI.
- Nous allons fabriquer un cadenas correspondant (Clé Publique) que nous allons installer sur le serveur de Sauvegarde.
- Ainsi, le serveur GLPI pourra entrer sans taper de code, juste en montrant sa clé.

Étape 1.1 : Générer les clés (Sur le serveur GLPI)

Attention : Comme notre script de sauvegarde tourne en **root** (via Cron), nous devons générer les clés pour l'utilisateur **root**.

1. Connecte-toi en root sur le serveur GLPI :

```
sudo -i
```

2. Génère la paire de clés (appuie sur **Entrée** à chaque question, ne mets **PAS** de passphrase, sinon l'automatisation est impossible) :

```
ssh-keygen -t rsa -b 4096
```

Le système a créé deux fichiers dans /root/.ssh/ : id_rsa (la clé) et id_rsa.pub (le cadenas).

Étape 1.2 : Envoyer le cadenas sur le serveur de sauvegarde

Toujours depuis le serveur GLPI, nous allons envoyer la clé publique vers le serveur B.

Remplace utilisateur par le nom de ton utilisateur sur le serveur B (ex: etudiant ou admin) et IP_SERVEUR_B par l'IP réelle.

```
ssh-copy-id utilisateur@IP_SERVEUR_B
```

(Tu devras taper le mot de passe de l'utilisateur distant une dernière fois pour installer la clé).

Étape 1.3 : Le test de vérité

C'est le moment critique. Toujours depuis le serveur GLPI (en root), tente de te connecter au serveur B :

```
ssh utilisateur@IP_SERVEUR_B
```

- **Si tu entres directement sans mot de passe :** C'est gagné ! Tape exit pour revenir sur GLPI.
- **Si on te demande un mot de passe :** Quelque chose a raté, recommence l'étape 1.2.

Partie 2 : Préparer le dossier de réception (Sur le Serveur B)

On ne va pas jeter les fichiers n'importe où.

Connecte-toi sur le Serveur B (Destination) et crée un dossier pour accueillir les archives.

```
mkdir -p /home/utilisateur/backups_glpi
```

(Remplace utilisateur par ton vrai nom d'utilisateur).

Partie 3 : Mise à jour du script de sauvegarde

Nous allons modifier notre script backup_glpi.sh (créé au TP précédent) pour ajouter l'étape d'envoi.

1. Sur le **Serveur GLPI**, édite le script :

```
nano /usr/local/bin/backup_glpi.sh
```

2. Ajoute la section suivante **à la fin du fichier**, juste avant le echo "Sauvegarde terminée..." :

```
# --- 4. EXTERNALISATION (Envoi vers Serveur B) ---
echo "Transfert vers le serveur de sauvegarde..."

# Configuration du serveur distant
REMOTE_USER="utilisateur"
REMOTE_IP="192.168.1.50"
REMOTE_DIR="/home/utilisateur/backups_glpi/"

# La commande RSYNC
# -a : archive (garde les droits et dates)
# -v : verbeux (affiche ce qu'il fait)
# -z : compresser pendant le transfert (pour aller plus vite)
# -e ssh : utiliser le tunnel SSH sécurisé

rsync -avz -e ssh $BACKUP_DIR/ $REMOTE_USER@$REMOTE_IP:$REMOTE_DIR

# Vérification du succès
if [ $? -eq 0 ]; then
    echo "Transfert réussi !"
else
    echo "ERREUR lors du transfert !"
fi
```

3. Enregistre (Ctrl+O) et quitte (Ctrl+X).

Partie 4 : Test final

Comme toujours, on teste manuellement avant d'attendre la nuit prochaine.

1. Lance le script sur le serveur GLPI :

```
/usr/local/bin/backup_glpi.sh
```

2. Observe les lignes qui défilent. Tu devrais voir :
    - La création du dump SQL.
    - La création du TAR.GZ.
    - **L'envoi des fichiers via rsync** (tu verras "sending incremental file list...").
3. Va vérifier sur le **Serveur B** :

```
ls -l /home/utilisateur/backups_glpi/
```

Si tes fichiers sont là, félicitations ! Tu as mis en place une stratégie de sauvegarde professionnelle et redondante.