Lycée Polyvalent Chevrollier

29 novembre 2024 | Version Final

Atelier de Professionnalisation

Écrit Individuelle #2 GSB

Élaboré par

Yann Duffay

Promotion

2 BTS SIO option SISR

Année scolaire

2024- 2025

TABLE DES MATIÈRES

I. Présentation globale du projet

L'objectif de ce projet est de mettre en place une architecture technique et des fonctions de sécurisation pour une application web destinée au suivi des comptes rendus et des frais de remboursement.

Cette application sera utilisée par les visiteurs, délégués et responsables de secteur, et sera accessible en ligne via Internet ou depuis les différents sites du réseau GSB.

Les principales exigences sont l'accessibilité restreinte aux acteurs de l'entreprise via une authentification préalable et le chiffrement des échanges entre les utilisateurs et le serveur.

L'architecture réseau devra inclure des périmètres de sécurité avec cloisonnement des réseaux, filtrage des accès publics, et sécurisation des bases de données.

Le projet doit aboutir à une solution technique fonctionnelle intégrée au réseau de l'entreprise, avec des serveurs sécurisés, en DMZ ou sur le LAN, et des mécanismes de protection comme le firewall et l'authentification.

II. Tâches effectuées

Dans le cadre de la mise en place des services SRV-GLPI et SRV-BACKUP, plusieurs opérations de recettage ont été réalisées. Ces tâches ont permis de valider la bonne configuration des serveurs, la sécurisation des accès HTTPS, ainsi que la sauvegarde des bases de données et des pages web.

2.1. SRV-GLPI

L'outil Gestion Libre de Parc Informatique va nous permettre de suivre et de gérer les actifs, incidents, demandes et maintenances au sein d'une infrastructure sur une interface web.

2.1.1. Configuration Serveur

Adresse IP/ Masque	Adresse Réseau	Passerelle	N° VLAN
192.168.70.4 /24	192.168.70.0	192.168.70.100	700

Dans un premier temps il nous faut télécharger une archive disponible sur github ou sur le site officiel ou contient les pages web qui composent l'application GLPI.

Version GLPI : 10.0.16

Une fois le fichier télécharger il faut veiller à ce que le fichier soit dans le répertoire de publication d'apache.

`tar -xvzf glpi-10.0.14.taz` va nous permettre d'extraire le fichier.

Maintenant il est nécessaire de créer un virtualhost qui va permette de faire cohabiter plusieurs serveurs web sur une même machine, dans notre cas cela nous sert à accéder à notre GLPI.

Étape 1 : Création d'un fichier glpi.conf

```
nano /etc/apache2/sites-available/glpi.conf
```

Notre serveur GLPI nécessite deux éléments essentiels pour fonctionner : une interface web et une base de données. Nous installons donc les paquets Apache2 et MariaDB-server.

Après l'installation d'Apache, nous configurons le serveur en téléchargeant la dernière version de GLPI et en créant un virtual host. Parallèlement, nous créons un utilisateur ainsi qu'une base de données pour GLPI.

Connexion à GLPI

Sélectionner notre BDD créé précédemment

2.1.2. Recettage HTTPS

Afin de sécuriser les échanges entre le client et le serveur nous allons mettre en place le HTTPS sur notre serveur Web pour éviter que les informations soit visible en clair.

Pour sécuriser les données, nous utilisons un **certificat SSL/TLS**, configuré dans le fichier `/etc/apache2/sites-available`.

Apache 2 propose deux sites par défaut : « **default** » et « **default-ssl** », pointant vers le répertoire « **/var/www** », avec le premier écoutant sur le port 80 (HTTP) et le second sur le port 443 (HTTPS).

Par défaut, seul le site « default » est activé, mais nous avons créé notre propre virtualhost c'est pourquoi nous désactivons le fichier default : `a2dissite default`

```
root@SRV-GLPI:/etc/apache2/sites-available# ls
000-default.conf  default-ssl.conf  glpi.conf
```

Contenu de notre VirtualHost :

```
<VirtualHost *:80>

        ServerName 192.168.70.4
        ServerAlias glpi


        DocumentRoot /var/www/glpi

        <Directory /var/www/glpi>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All
                Order allow,deny
                allow from all
                AuthType Basic
        </Directory>

        LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{Useragent}i\"" combined
        CustomLog ${APACHE_LOG_DIR}/glpi_access.log combined
        ErrorLog ${APACHE_LOG_DIR}/glpi_error.log

</VirtualHost>
```

Le site SSL est déjà préconfiguré mais pas adapté à notre cas car nous avons créé un dossier spécial GLPI nous devons donc modifier le chemin d'accès de nos pages.

```
root@SRV-GLPI:/var/www# ls
glpi  html
```

Maintenant il suffit donc d'activer le module SSL et le site « default-ssl » avec les commandes suivantes :

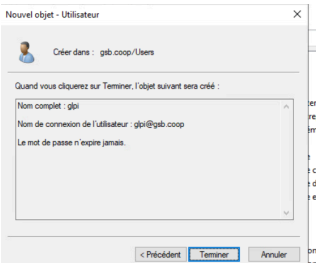
```
a2enmod ssl
a2ensite default-ssl
Systemctl restart apache2
```

Si nous souhaitons vérifier de manière approfondie son fonctionnement, nous pourrions installer un Wireshark sur une machine cliente et lancer une requête vers le serveur GLPI en constatant que le port utilisé 443 et que les informations ne circulent pas en clair.

2.1.3. LDAP

LDAP va nous permettre de lier notre GLPI à notre AD, ce qui va nous permettre de récolter les données tels que le nom, prénom, adresse mail au sein de notre GLPI ou on pourra distinguer l'ensemble de nos utilisateurs.

Nous allons dans un premier temps créer un utilisateur **glpi** sur notre AD en renseignant à quel groupe il appartient.



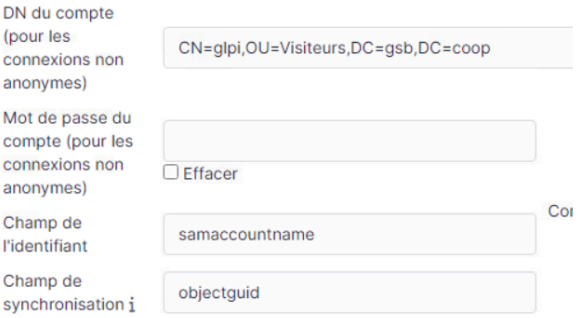
Une fois créé, on teste ce nouvel utilisateur sur une de nos machines test afin de vérifier son bon fonctionnement.

A présent sur l'interface web de GLPI, on se rend **Annuaire LDAP**, on clique sur Active Directory qui va permettre de générer automatiquement nos informations du serveur (IP, Nom, Filtre de connexion).

Concernant l'option **Actif** on change et on clique sur **Oui** comme ci-dessus.

Pour **BaseDN**, on renseigne notre OU, le contrôleur de domaine avec son extension.

Pour remplir **DN**, on se rend dans Outils > Modification ADSI et on cherche notre correspondant, dans notre cas glpi. On copie le nom unique.



On renseigne le mot de passe de notre compte glpi créé précédemment, puis on ajoute

Une fois les paramètres de notre Active Directory renseignés on va tester son bon fonctionnement.

Si le test est réussi, il sera affiché.

2.1.3.3. Importation des utilisateurs

Administration ➡ Utilisateurs ➡ Liaison annuaire LDAP

On sélectionne

On clique sur Rechercher

Les utilisateurs s'affichent en dessous ce qui nous permet de vérifier que nos utilisateurs remontent correctement.

2.2. SRV-BACKUP

Nous mettons en place un serveur de sauvegarde afin de garantir la sécurité et l'intégrité de nos données en cas de pertes, erreurs ou attaques.

2.2.1. Configuration Serveur

Adresse IP/ Masque	Adresse Réseau	Passerelle	N° VLAN
192.168.75.1 /24	192.168.75.0	192.168.75.100	705

Pour accueillir les **différentes sauvegardes**, nous faisons le choix de créer un dossier spécifique pour les différentes sauvegardes (**BDD GSB, BDD GLPI, Pages Web**).

```
root@SRV-BACKUP:/home/sauvegarde# ls
BDD_GLPI  BDD_GSB  PAGES_WEB
```

Sur chacun des serveurs hébergement les données que nous souhaitons sauvegarder, nous installons le paquet `sshpass` afin d'éviter de devoir s'authentifier manuellement.

Dans notre cas, nous choisissons de faire une **sauvegarde complète tous les jours à 23h** car nous savons qu'à ce moment personne n'est connecté, explicitement renseigné dans la **charte informatique** que chaque utilisateur à signé.

Optionnel : Installation du paquet `tree` pour me visualiser l'arborescence.

2.2.2. Recettage Script Sauvegarde BDD-GSB

Sur notre SRV-BDD, nous créons un fichier **script_bdd-gsb.sh** ou nous allons retrouver le contenu de notre script.

Ci-dessous le script de la sauvegarde de la base de données GSB :

```
#!/bin/bash

# Variables pour BDD
USER="root"
PASSWORD="root"
DATABASE="GSB_BDD"
DATE=$(date +%Y%m%d_%Hh%M) BACKUP_FILE="GSB_BDD-${DATE}.sql"

# Variables transfère via sshpass
SSH_USER="root"
SSH_PASSWORD="root"

# Sauvegarde et Envoie vers SRV-BACKUP
mysqldump -u ${USER} -p${PASSWORD} ${DATABASE} | sshpass -p "${SSH_PASSWORD}"
ssh ${SSH_USER} @192.168.75.1 "cat > /home/sauvegarde/BDD_GSB/${BACKUP_FILE}"
```

Pour des raisons d'identification, nous avons fait en sorte que chaque sauvegarde soit nommée par la **date, les heures et minutes** facilitant la gestion pour rechercher une sauvegarde à une date précise.

Maintenant que le script à été créé il nous faut l'automatiser, c'est pourquoi nous allons utiliser **cron** disponible nativement sur debian.

`crontab -e` pour modifier le fichier

```
0 23 * * * /home/script_bdd-gsb.sh
```

Nous définissons l'heure, la fréquence, et on spécifie le chemin du script qui doit s'exécuter.

Il nous suffit de vérifier maintenant sur notre SRV-BACKUP que les sauvegardes ont bien été exécutées sans problème, en vérifiant aussi le contenu avec un `cat NomFichier`.

```
root@SRV-BACKUP:/home/sauvegarde/BDD_GSB# ls
GSB_BDD-20241011_23h00.sql  GSB_BDD-20241013_23h00.sql  GSB_BDD-20241016_23h00.sql
GSB_BDD-20241012_23h00.sql  GSB_BDD-20241014_23h00.sql
GSB_BDD-20241013_18h40.sql  GSB_BDD-20241015_23h00.sql
```

2.2.3. Recettage Script Sauvegarde BDD-GLPI

Sur notre SRV-GLPI, nous créons un fichier **script_bdd-glpi.sh** ou nous allons retrouver le contenu de notre script.

Ci-dessous le script de la sauvegarde de la base de données GLPI:

```
#!/bin/bash

# Variables pour BDD
USER="root"
PASSWORD="root"
DATABASE="GLPI_BDD"
DATE=$(date +%Y%m%d_%Hh%M)
BACKUP_FILE="GLPI_BDD-${DATE}.sql"

# Variables transfère via sshpass
SSH_USER="root"
SSH_PASSWORD="root"

# Sauvegarde et Envoie vers SRV-BACKUP
mysqldump -u ${USER} -p${PASSWORD} ${DATABASE} | sshpass -p "${SSH_PASSWORD}"
ssh ${SSH_USER} @192.168.75.1 "cat >
/home/sauvegarde/BDD_GLPI/${BACKUP_FILE}"
```

2.2.4. Recettage Script Sauvegarde Pages Web

Sur notre SRV-WEB, nous créons un fichier **script_backup-pages.sh** ou nous allons retrouver le contenu de notre script.

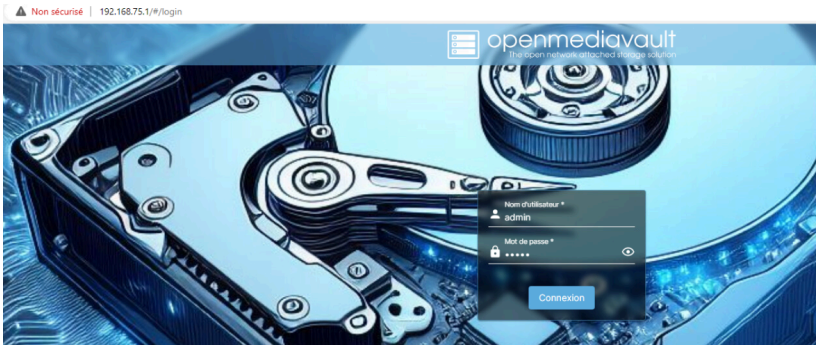
Ci-dessous le script de la sauvegarde des pages web :

```
#!/bin/bash

# Variables
DATE=$(date +%Y%m%d_%Hh%M)
BACKUP_FILE="PagesWeb-{ $DATE}"

# Variables transfère via sshpass
SSH_USER="root"
SSH_PASSWORD="root"

# Création Dossier destination
sshpass -p "${SSH_PASSWORD}" ssh ${SSH_USER} @192.168.75.1 "mkdir -p
/home/sauvegarde/PAGES_WEB/${BACKUP_FILE}"

# Envoie vers SRV-BACKUP
```

```
sshpass -p "${SSH_PASSWORD}" scp -r /var/www/* ${SSH_USER} @192.168.75.1:
/home/sauvegarde/PAGES_WEB/${BACKUP_FILE}
```

2.2.5. Recettage OpenMediaVault

Après réflexion pour la solution préalablement choisie pour le serveur de sauvegarde, nous mettons en place OpenMediaVault. Cette outil va nous permettre d'effectuer nos sauvegardes de manière sécurisé et avoir un outil de visualisation.



III. Veille Technologique

Afin de faire face au problème de sécurité des scripts de sauvegardes, des recherches nous ont permis de visualiser différentes solutions possibles comme TrueNas ou OpenMediaVault. Cela nous permet de gérer via une interface web et d'avoir un point de recul sur l'espace de stockage pour déterminer l'utilisation et l'adpater.

IV. Synthèse de groupe

Nous avons continué l'installation de divers serveurs. Pour l'instant, nous avons pu configurer :

- Un serveur AD – DNS| Windows Server |
- Un serveur de supervision CENTREON | DEBIAN12 |
- Un serveur GLPI LDAP | DEBIAN 12 |
- Un routeur pare-feu PfSense
- Un serveur de backup Solution OMV | DEBIAN 12 |
- Un serveur de base de données connecté à l'intranet | DEBIAN 12 |
- Un serveur de partage de fichiers VSFTPD (Filezilla)| DEBIAN 12 |
- Un serveur web pour l'intranet Apache2 | DEBIAN 12 |
- Un serveur de temps NTP, PfSense | DEBIAN 12 |

Nous projetons l'installation d'autres serveurs :

- Un serveur Proxy | DEBIAN 12 |
- Un bastion (organisation de projets) | DEBIAN 12 |
- Un serveur de VPN OpenVPN (PfSense)
- Un DHCP sur la partie physique
- Nous réfléchissons sur l'utilité d'un serveur IDS / IPS.

Nous sommes restés organisés tout au long de notre projet grâce à Trello. Cette méthode nous aura permis de maintenir une organisation fiable. De plus, nous avons décidé de faire un point à chaque début de séance afin de communiquer sur le projet, savoir ce que l'autre va faire lors de sa séance...

L'avancée de notre projet nous semble satisfaisante. Nous pensons être dans les temps.

Pour ce qui est de la cybersécurité, nous y prêtons beaucoup d'attention. En effet, nous essayons au maximum de penser d'abord par la sécurité. Cela va du passage des serveurs web en HTTPS jusqu'au changements de ports de communication en passant par des serveurs de défense et de prévention.