

PREUVES DE RÉALISATION — STAGE BTS SIO

Mission : Agent IA de supervision pour Datacenter

Entreprise : LOXYS Solutions Informatiques

Preuve 1 — Mise en place des niveaux d'habilitation Zabbix

1.1 Problème identifié

L'utilisateur flowise-bot avait le rôle 'User role' avec des permissions 'None' sur tous les groupes d'hôtes. Zabbix bloquait silencieusement l'envoi des notifications sans erreur visible.

1.2 Capture d'écran — Permissions initiales (None)

Extrait de la conversation du diagnostic :

Group	Type	Permissions	Statut
All groups	Hosts	None 	AVANT correction
Databases	Hosts	Read 	APRÈS correction
LOXYS Servers	Hosts	Read 	APRÈS correction
Zabbix servers	Hosts	Read 	APRÈS correction

1.3 Actions réalisées

- Navigation dans Users > User groups > Zabbix administrators > Host permissions
- Ajout d'un accès Read sur les groupes Databases, LOXYS Servers et Zabbix servers
- Vérification que flowise-bot (User role) hérite bien des permissions du groupe

1.4 Résultat

Après correction, le Action log de Zabbix affichait bien les tentatives d'envoi (Sent) au lieu de l'absence totale de log constatée auparavant. L'alerte était désormais transmise à l'agent Flowise.

Preuve 2 — Création d'un compte SSH restreint (flowise-diag)

2.1 Objectif

Permettre à l'agent IA Flowise de se connecter au serveur Database pour collecter des données de diagnostic en temps réel, sans lui donner de droits d'administration ni de shell interactif.

2.2 Commandes exécutées

Création de l'utilisateur dédié sur le serveur Database (192.168.1.111) :

```
sudo useradd -m -s /bin/bash flowise-diag
sudo mkdir -p /home/flowise-diag/.ssh
```

```
sudo chmod 700 /home/flowise-diag/.ssh
```

Génération d'une clé SSH Ed25519 dédiée sur le serveur Flowise :

```
ssh-keygen -t ed25519 -f ~/.ssh/flowise_diag_key -C "flowise-diag" -N ""
```

Configuration de authorized_keys avec commande forcée (restriction maximale) :

```
command="/usr/local/bin/flowise-diag.sh",no-port-forwarding,  
no-X11-forwarding,no-agent-forwarding,no-pty ssh-ed25519 AAAA...
```

2.3 Script de diagnostic en lecture seule

Contenu de /usr/local/bin/flowise-diag.sh (aucune commande d'écriture ou de suppression) :

```
#!/bin/bash  
echo "--- CPU ---"; top -bn1 | head -20  
echo "--- MEM ---"; free -h  
echo "--- DISK ---"; df -h  
echo "--- TOP PROCESSES (CPU) ---"; ps aux --sort=-%cpu | head -10  
echo "--- TOP PROCESSES (MEM) ---"; ps aux --sort=-%mem | head -10  
echo "--- CONNEXIONS RESEAU ---"; ss -tunp 2>/dev/null | head -20  
echo "--- LOGS ERREURS ---"; journalctl -p err -n 30 --no-pager
```

2.4 Mesures de sécurité appliquées

Mesure	Détail
Aucun droit sudo	Aucun fichier dans /etc/sudoers.d/
Commande forcée SSH	authorized_keys avec command="..."
Pas de shell interactif	no-pty dans authorized_keys
Lecture seule uniquement	Script sans écriture/suppression
Accès logs système	sudo usermod -aG adm flowise-diag
Clé Ed25519 dédiée	Clé SSH distincte de celle de l'admin

2.5 Test de validation

Test effectué depuis le serveur Flowise vers Database :

```
ssh -i ~/.ssh/flowise_diag_key flowise-diag@192.168.1.111
```

Résultat obtenu : le script s'exécute automatiquement, retourne les données système, puis la connexion se ferme sans shell interactif. Aucune commande arbitraire n'est possible.

Preuve 3 — Résultat final : mail d'alerte avec diagnostic réel

3.1 Flux complet mis en place

Ét ap e	Action	Résultat
1	Zabbix détecte CPU > 90%	Trigger last() déclenché en < 15s
2	Webhook appelle l'agent Flowise	Timeout porté à 60s
3	Agent appelle diagnostic_ssh_database	Données réelles collectées en SSH
4	Agent génère le diagnostic	Vrais PIDs identifiés (stress-ng-cpu)
5	Mail envoyé au format LOXYS	Diagnostic + marche à suivre reçus

3.2 Extrait du mail reçu lors du stress test

ALERTE WARNING — Database

Problème : Linux: High CPU utilization (over 90% for 5m)

Valeur : 97.68 %

DIAGNOSTIC PRINCIPAL : Les processus stress-ng-cpu consomment une grande quantité de ressources CPU. [données réelles SSH]

3.3 Lien avec les critères BTS SIO E4

Critère E4	Réalisation
Mettre en place les niveaux d'habilitation	Permissions Zabbix + compte SSH restreint
Vérifier la continuité d'un service	Alertes automatiques + persistance Docker
Réaliser les tests d'intégration	Stress tests CPU, tests webhook, curl API
Déployer un service	Déploiement complet Zabbix + Flowise + IA
Traiter des demandes réseau et système	Résolution triggers, webhooks, SSH, Docker

Document généré à partir des échanges de travail réels durant le stage