Documentation

Projet de Lab Réseau & Sécurité

1️⃣. Objectif général du projet

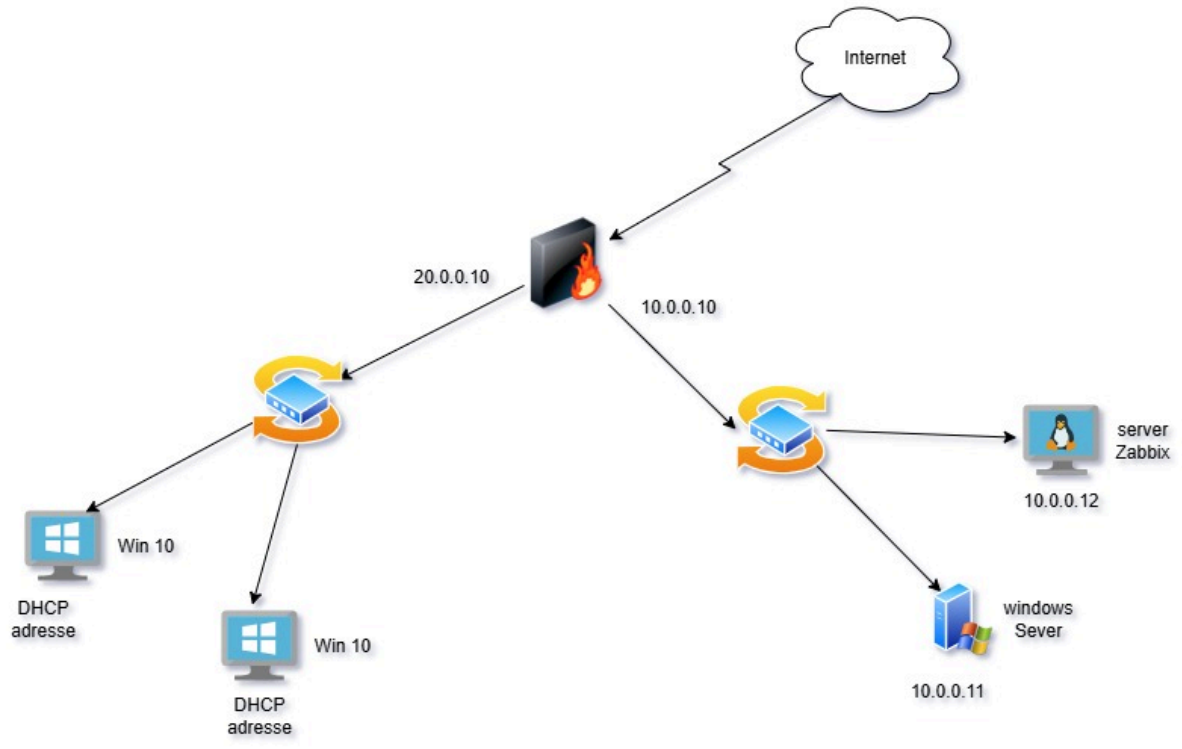
Mettre en place une infrastructure réseau sécurisée et supervisée pour une PME fictive, avec :

- un **contrôleur de domaine Windows Server (AD, DNS, DHCP)**,
- un **pare-feu pfSense** pour la sécurité et le routage inter-VLAN,
- des **postes clients Windows** simulant les employés,
- et une **solution de supervision Zabbix** pour le monitoring global du réseau.

  But pédagogique : comprendre et administrer une architecture d'entreprise complète intégrant sécurité, services réseau et supervision.

2️⃣. Architecture du réseau

a. Schéma logique



b. Description des sous-réseaux

Réseau	Adresse	Rôle	Appareils connectés
LAN server	10.0.0.0/24	Infrastructure interne	Windows Server, Zabbix
LAN Utilisateurs	20.0.0.0/24	Postes clients	PC Windows 10 (DHCP)
WAN	DHCP (fournisseur internet simulé)	Accès Internet	pfSense

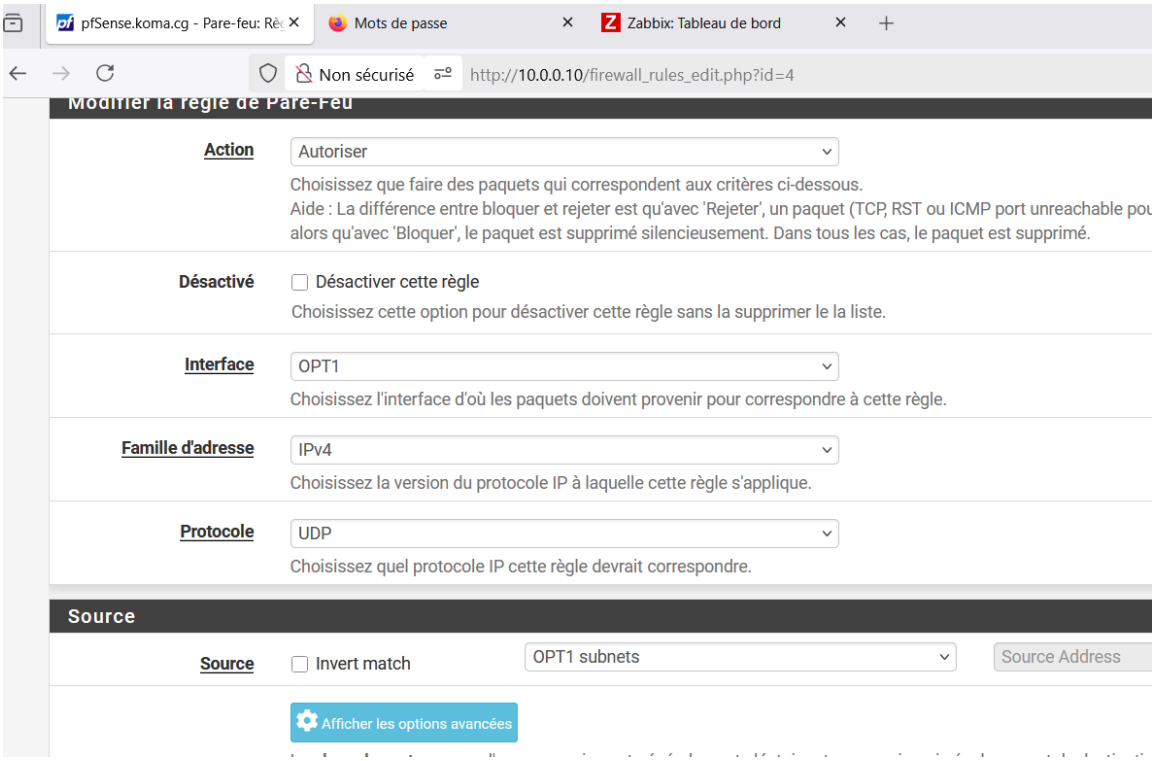
3️⃣. Composants du lab

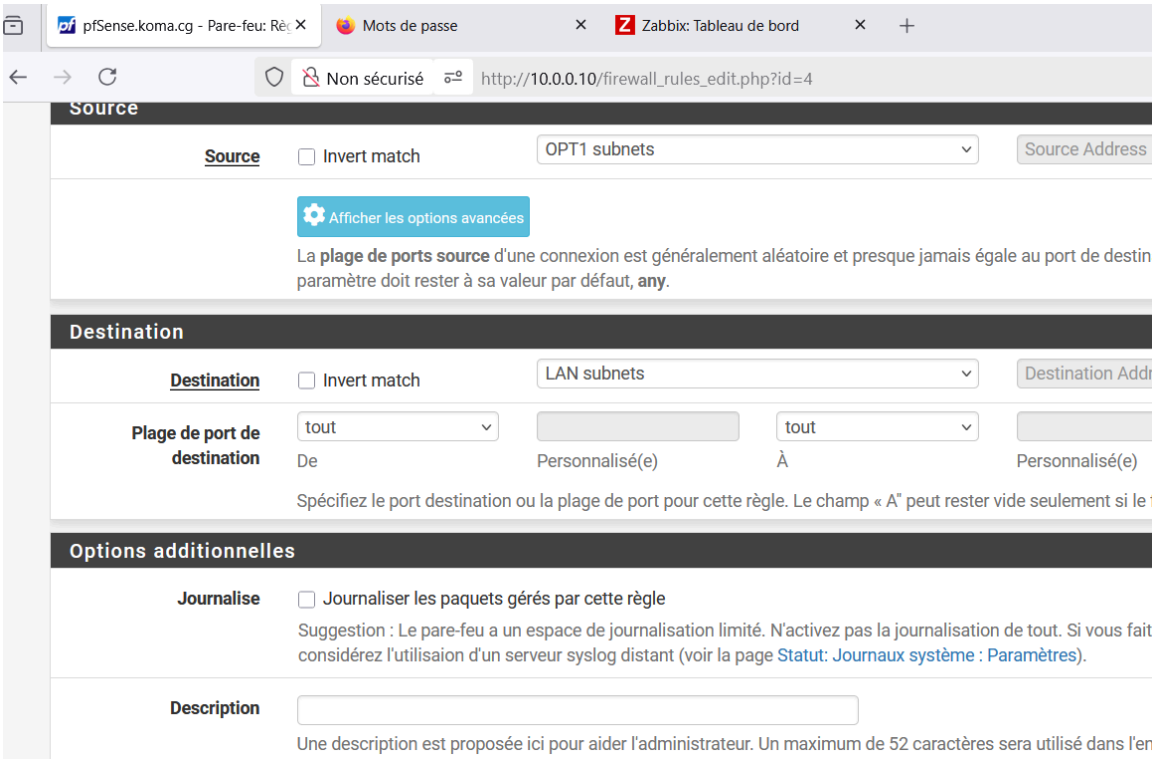
Composant	Rôle	Système	IP / Interfaces
pfSense	Pare-feu et routage	pfSense 2.7.x	WAN : 20.0.0.10 / LAN : 10.0.0.10
Windows Server 2022	Active Directory, DNS, DHCP	Windows Server	10.0.0.11
Zabbix Server	Supervision et monitoring	Ubuntu 24.04 + Zabbix 6.x	10.0.0.12
Clients Windows 10	Postes utilisateurs	Windows 10	DHCP (20.0.0.x)

4️⃣. Configuration technique

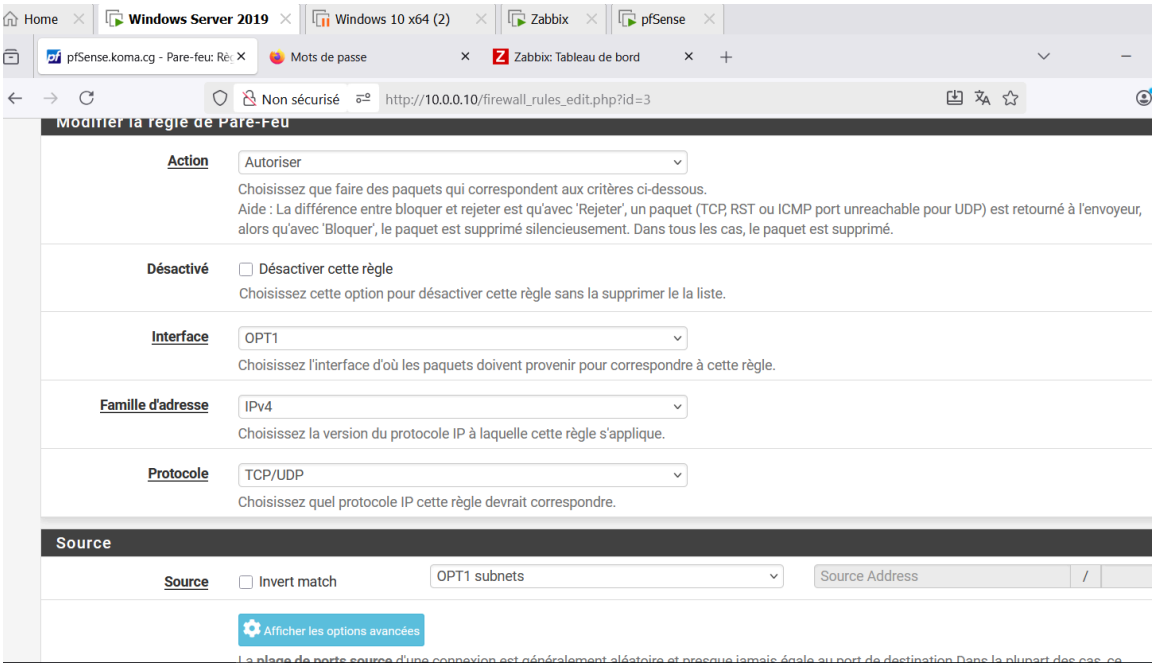
a. pfSense

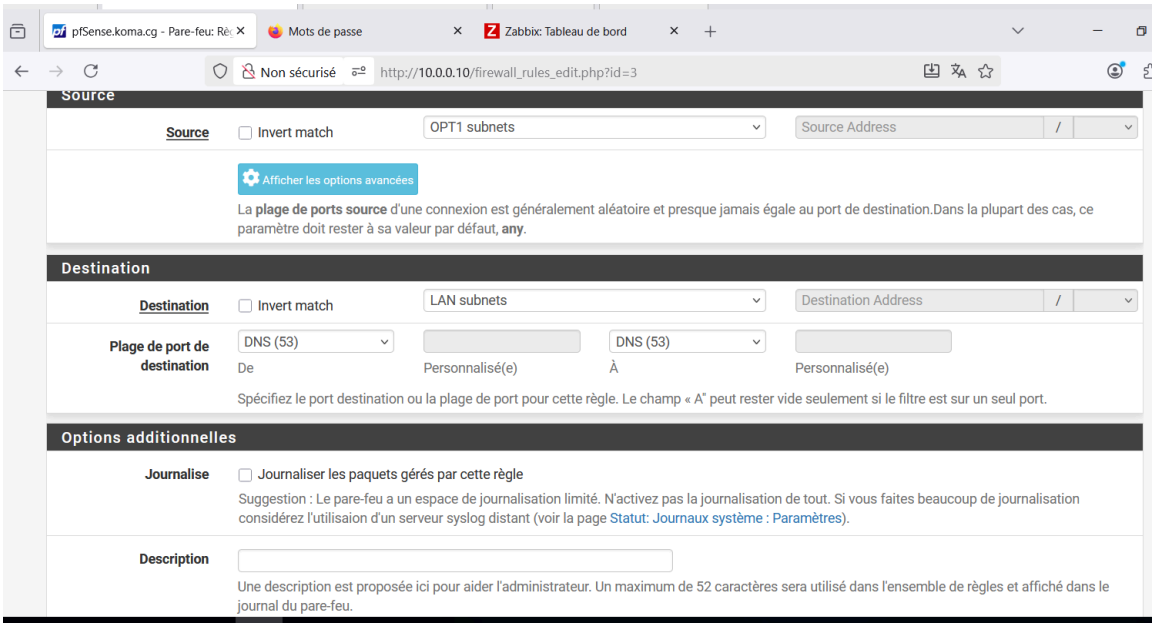
- Interface WAN : (DHCP / Internet)
- Interface LAN : 10.0.0.10 (Gateway du réseau interne)
- Interface OPT1 : 20.0.0.10
- Règles :
  - Autoriser trafic UDP entre OPT1→ LAN Afin que nos machines clientes du côté OPT1 puisse avoir une adresse DHCP géré par le server Windows (Ne pas oublié de configurer le DHCP relay)





- Autoriser le trafic DNS entre win 10 et win ser





  - Bloquer accès non autorisé WAN → LAN
  - Zabbix Agent installé pour la supervision depuis System/Package_Manager/Available_Packages (faire attention à la version téléchargé elle doit être la même que celle de notre Zabbix)
  - Rediriger les logs vers le serveur Zabbix
- Services : DHCP désactivé (géré par Windows Server)

▼ NB: Principe de création de règle

🔹 1. Principe général du pare-feu pfSense

pfSense est **stateful**, c'est-à-dire qu'il **suit les connexions** :

- Si tu autorises une connexion sortante, la réponse entrante sera automatiquement autorisée.
- Donc tu n'as pas besoin d'écrire deux règles (une pour chaque sens), sauf dans des cas spécifiques (comme pour du NAT ou des serveurs exposés).

🔹 2. Sens d'application des règles

➤ Les règles s'appliquent sur une interface donnée, dans le sens "entrant" (IN)

👉 Cela veut dire :

- Une règle sur l'interface **LAN** s'applique **aux paquets qui entrent dans pfSense depuis le LAN**.
- Une règle sur l'interface **WAN** s'applique **aux paquets qui arrivent depuis Internet vers pfSense**.

> 💡 Important : le trafic sortant du LAN vers Internet "entre" sur l'interface LAN.
>
> Le trafic entrant depuis Internet "entre" sur l'interface WAN.

🔹 3. Ordre de lecture des règles

pfSense **lit les règles de haut en bas**, et **s'arrête à la première qui correspond**.

- Si aucune règle ne correspond → le trafic est **bloqué par défaut** (policy deny).
- Tu peux réordonner les règles par glisser-déposer.

b. Windows Server

- AD DS installé → Domaine : *koma.cg*
- DNS intégré à AD

- DHCP configuré pour le réseau **20.0.0.0/24**
- Comptes utilisateurs et groupes créés
- Zabbix Agent installé pour la supervision (il peut être téléchargé en recherchant Zabbix agent dans le navigateur, s'assurer la version télécharger est la plus proche de la version de Zabbix)

c. Serveur Zabbix (Ubuntu)

- Installation :
  - ▼ Suivre les étapes suivantes

    sudo dnf update & sudo dnf upgrade -y

    sudo dnf install httpd -y
    sudo systemctl start httpd
    sudo systemctl enable httpd

    sudo firewall-cmd --permanent --zone=public --add-service=http
    sudo firewall-cmd --permanent --zone=public --add-service=https
    sudo firewall-cmd --reload

    sudo dnf install mysql-server -y
    sudo systemctl start mysqld.service
    sudo systemctl enable mysqld.service
    sudo systemctl status mysqld.service
    sudo mysql_secure_installation

    sudo rpm -Uvh https://repo.zabbix.com/zabbix/6.4/rhel/9/x86_64/zabbix-release-6.4-1.el9.noarch.rpm

    sudo dnf install zabbix-server-mysql zabbix-web-mysql zabbix-apache-conf zabbix-sql-scripts zabbix-selinux-policy zabbix-agent

    mysql -uroot -p
    create database zabbix character set utf8mb4 collate utf8mb4_bin;
    create user zabbix@localhost identified by 'password';
    grant all privileges on zabbix.* to zabbix@localhost;
    set global log_bin_trust_function_creators = 1;
    quit;

    sudo zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -uzabbix -p zabbix

mysql -uroot -p
set global log_bin_trust_function_creators = 0;
quit;

sudo systemctl restart zabbix-server zabbix-agent httpd php-fpm
sudo systemctl enable zabbix-server zabbix-agent httpd php-fpm

sudo firewall-cmd --zone=public --permanent --add-port=10050/tcp
sudo firewall-cmd --zone=public --permanent --add-port=10051/tcp
sudo firewall-cmd --reload

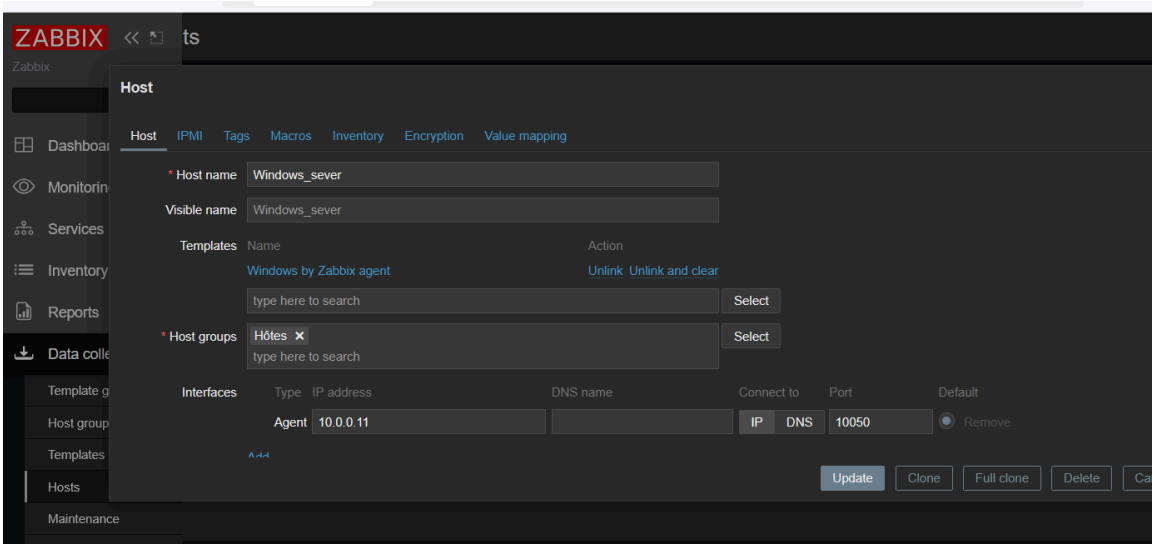
FEE07426-2359-46CF-BE9B-AC24DD54BFE1.pdf

- Base de données MySQL créée pour Zabbix

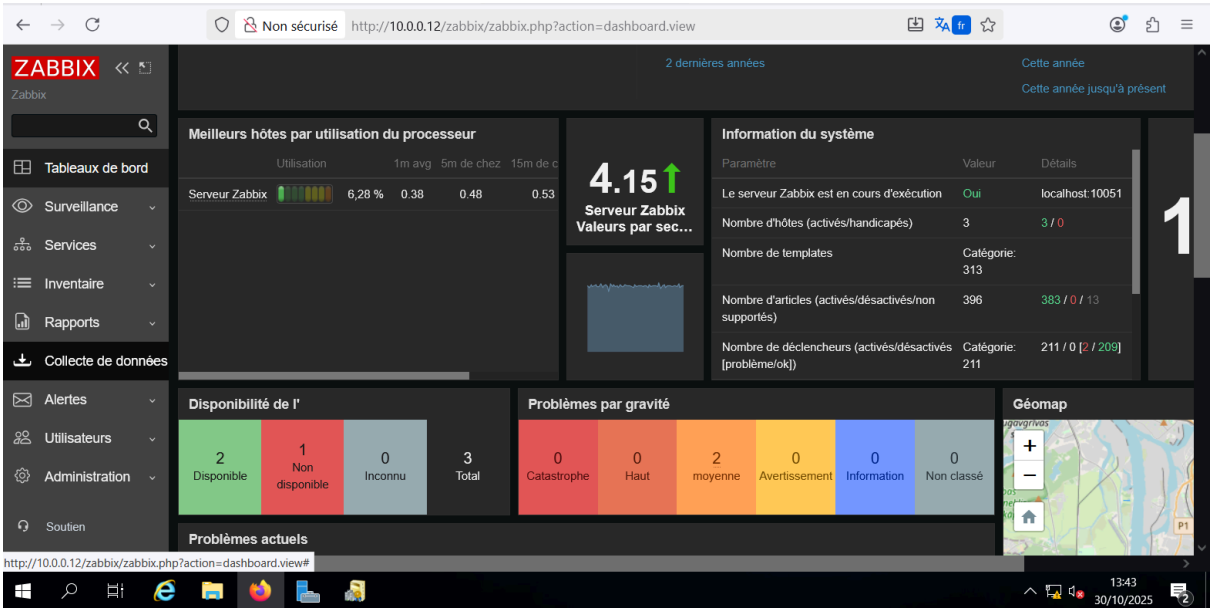
▼ Agents installés sur les postes et serveurs du lab

Ajoute la machine sur l'interface Zabbix

1. Connecte-toi à **Zabbix Web** → *Configuration* → *Hosts*
2. Clique sur **Create host**
3. Remplis :
   - **Host name** → même que dans `Hostname` du client
   - **Visible name** → un nom lisible (ex. Poste Win10)
   - **Groups** → "Linux servers" ou "Windows hosts"
   - **Interfaces** →
     - Type : **Agent**
     - IP : adresse IP du client (ex. 10.0.0.15)
     - Port : **10050**
4. Onglet **Templates** → **Link new templates**
   - Pour Windows : `Template OS Windows by Zabbix agent`
   - Pour Linux : `Template OS Linux by Zabbix agent`
5. Clique sur **Add**



- Tableaux de bord configurés pour supervision réseau et CPU/RAM



5️⃣. Tests de fonctionnement

Test	Description	Résultat attendu
🧭 **Ping inter-VLAN**	Ping entre 10.0.0.x ↔ 20.0.0.x via pfSense	✅ Réussi
🌐 **Navigation Internet**	Clients accèdent au web via pfSense	✅ OK
🧱 **Filtrage pfSense**	Blocage d'un port spécifique (ex : 23/Telnet)	✅ Fonctionnel
📡 **Supervision Zabbix**	Surveillance CPU, RAM, ping, uptime de tous les hôtes	✅ Données remontées

Test	Description	Résultat attendu
👥 **Connexion domaine AD**	Les clients Windows rejoignent le domaine `entreprise.local`	✅ OK

6️⃣. Supervision et sécurité

- Zabbix surveille :
  - disponibilité du pare-feu, du serveur AD, du DNS, et des clients
  - charge CPU/mémoire des machines
- Alertes configurées pour :
  - perte de ping, CPU > 80%, espace disque < 10%
- pfSense envoie ses logs vers Zabbix (ou un serveur Syslog distant)
- Comptes AD limités, stratégies de mot de passe appliquées

9️⃣. Conclusion

🎯 **Bilan du projet :**

Ce lab m'a permis de :

- comprendre la configuration d'un pare-feu pfSense,
- déployer et gérer un domaine Active Directory,
- superviser un réseau avec Zabbix,
- appliquer des concepts de sécurité réseau et de segmentation.

💡 **Perspectives d'amélioration :**

- Ajouter un serveur mail interne, un proxy Squid ou un IDS/IPS (Suricata / Snort)
- Implémenter un VPN (OpenVPN ou WireGuard)
- Automatiser la configuration via Ansible