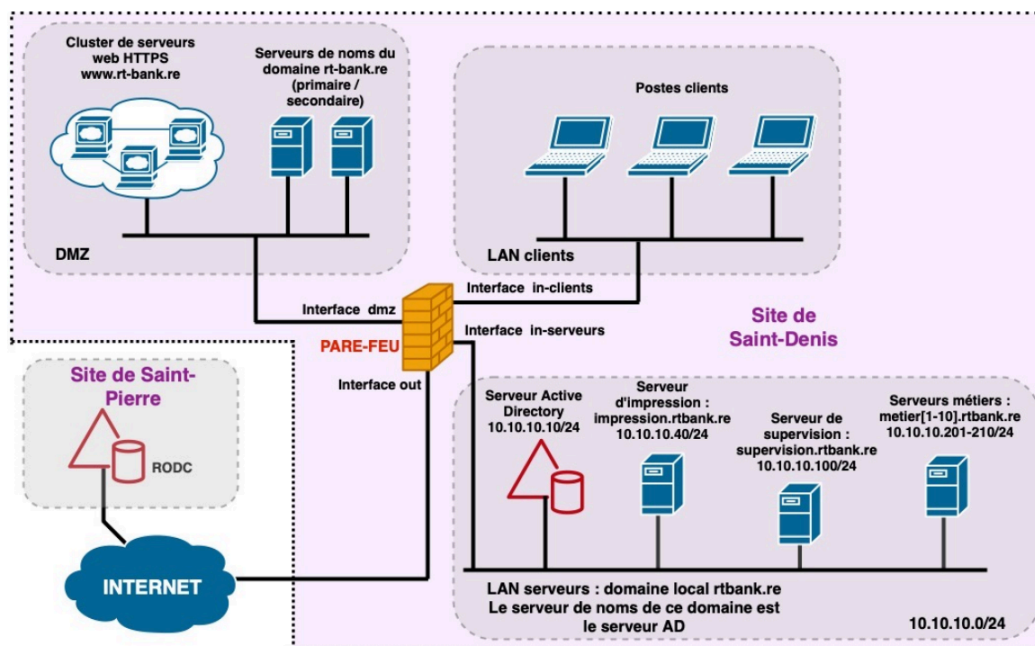


Institut Universitaire Technologique  
réseaux et télécoms  
Cybersécurité

*Projet en SAE Cybersécurité :*

# Concevoir un réseau sécurisé virtuel multis sites



**Réalisé par :**

HONORINE Kylian  
GRONDIN  
GROUFFAULD Joël  
GRONDIN Benjamin  
Tommy

**Encadré par :**

SITAL DAHONE Aloïs  
Angélique  
  
MONTEGU  
  
ARNASSALOM Kevin

# TABLE DES MATIÈRES

<b>1. Objectif du projet.....</b>	<b>3</b>
<b>3. Organisation du travail.....</b>	<b>3</b>
<b>4. Infrastructure demandée.....</b>	<b>4</b>
4.1 Plan réseau.....	4
4.2 Services à mettre en place.....	8
<b>5. Points techniques majeurs.....</b>	<b>9</b>
Tâche 3 : Serveur DNS.....	17
Tâche 4 : Active Directory.....	22

6. Méthodologie de sécurité appliquée.....	35
7. Conclusion.....	39

## **1. Objectif du projet**

Mettre en œuvre une infrastructure réseau **sécurisée** pour une banque fictive, **RT Bank**, répartie sur deux sites (Saint-Denis – Siège / Saint-Pierre – Succursale), en respectant un **cahier des charges** précis et en appliquant les bonnes pratiques de **cybersécurité**.

## **2. Compétences visées**

- AC24.01Cyber : Bonnes pratiques de cybersécurité
- AC24.02Cyber : Sécurisation d'infrastructure réseau
- AC24.03Cyber : Sécurisation des services
- AC24.04Cyber : Usage des outils cryptographiques
- AC24.05Cyber : Connaissance des types d'attaques
- AC25.01Cyber : Administration anti-malwares
- AC25.02Cyber : Utilisation d'outils de pentest

## **3. Organisation du travail**

- **Groupes** : binômes (2 étudiants, exceptionnellement 3)
- **Déroulement** :
  - 19h de cours/TD, 12h de TP encadrés
  - 50h de projet (16h encadrées + 34h en autonomie)
- **Durée** : du 13 mars 2023 au 18 avril 2023
- **Livrables** :
  - Contrôles intermédiaires (QCM/CR TP)
  - Soutenance/démonstration fonctionnelle

## **4. Infrastructure demandée**

### **4.1 Plan réseau**

- **Pare-feu** (Stormshield, PFSense ou OPNSense) avec **4 interfaces** :
  - DMZ (serveurs publics)
  - LAN Clients (réseaux utilisateurs internes)
  - LAN Serveurs (réseaux des serveurs internes)
  - WAN (sortie Internet)
- **Plages IP** :

- LAN-Serveurs : 192.168.130.1/23
- LAN-Clients / DMZ : plages privées au choix selon RFC1918
- Internet : Attribution d'IP statiques (ex: 192.168.41.X pour filaire)

## 4.1 Architecture Réseau

Pour répondre au besoin d'une infrastructure multisites sécurisée, nous avons mis en œuvre un réseau segmenté à l'aide d'un pare-feu à quatre interfaces, déployé sous Stormshield EVA. Ce découpage répond aux exigences de cloisonnement entre les différentes zones fonctionnelles et s'inspire des modèles de sécurité professionnels.

### 1. Interfaces du pare-feu et leur rôle

Interface DMZ (demilitarized zone) :

C'est une zone tampon entre Internet et le réseau interne, conçue pour héberger les services exposés au public, tout en les isolant du LAN.

Nous y avons placé :

Les serveurs Web Apache (accessibles via HTTPS)

Le serveur DNS primaire pour le domaine public rt-bank.re

### Interface LAN-Clients :

Ce réseau contient les postes de travail des utilisateurs (clients Windows 10).

Il ne doit en aucun cas être directement exposé à Internet.

Les accès Internet ou aux services (Web, AD, fichiers) passent uniquement via le pare-feu.

### Interface LAN-Serveurs :

Ce réseau est réservé aux serveurs internes critiques, incluant :

Le contrôleur principal Active Directory

Le serveur DNS secondaire

Le serveur de supervision Zabbix

D'éventuels services de fichiers ou d'impression internes

### Interface WAN :

Reliée à l'extérieur (réseau pédagogique de l'IUT), cette interface donne accès à Internet via NAT. Elle est également utilisée pour publier les services

externes de la DMZ (Web, DNS), via une traduction d'adresses statique (NAT statique).

## 2. Plan d'adressage IP

L'adressage a été soigneusement choisi pour garantir la lisibilité, l'évolutivité et l'isolation entre les réseaux :

LAN-Serveurs :

Réseau réservé : 192.168.130.1/23

» Un sous-réseau dédié aux serveurs internes. Ce choix garantit un repérage rapide des machines critiques.

Ex : 10.10.10.1 (AD), 10.10.10.2 (DNS secondaire), 10.10.10.3 (Zabbix)

DMZ / LAN-Clients :

Plages d'adresses issues du RFC1918 (réseaux privés standards : 192.168.x.x, 10.x.x.x ou 172.16.x.x).

» Cela permet d'éviter tout chevauchement avec les adresses Internet publiques.

Ex :

DMZ : 192.168.20.0/24 » serveurs exposés

LAN-Clients : 192.168.120.0/23 » machines utilisateur

WAN (Internet) :

Réseau fourni par l'IUT de La Réunion pour les projets pédagogiques :

Filaire : 192.168.41.x/23

Wi-Fi : 192.168.51.x/23

Chaque étudiant utilise un bloc de 5 adresses statiques à partir de son numéro dans la liste d'émargement.

Exemple : étudiant n°13 » 192.168.41.65 à 69

Ces adresses sont utilisées pour le NAT statique, permettant de rendre le serveur Web et DNS accessibles depuis Internet (en toute sécurité via le pare-feu).

### 3. Séparation des flux & logique de sécurité



Chaque réseau est isolé logiquement par le pare-feu, et seuls les flux nécessaires sont autorisés :

Les clients LAN peuvent interroger le serveur AD ou naviguer via NAT

Les serveurs DMZ peuvent répondre aux requêtes DNS/Web mais n'ont pas accès aux serveurs internes

Les flux entrant depuis Internet sont filtrés strictement (ex : seule la publication du site Web est autorisée)

Aucune communication directe entre la DMZ et le LAN-Serveurs n'est autorisée

## 4.2 Services à mettre en place

Tâche	Description
-------	-------------

<b>Pare-feu + DMZ</b>	Installer, configurer filtrage, NAT statique
<b>Serveur Web sécurisé</b>	2 serveurs Apache + HAProxy pour répartition de charge + HTTPS (TLS + Certificats)
<b>Serveur DNS (rt-bank.re)</b>	DNS primaire + secondaire sécurisés (DMZ)
<b>Active Directory</b>	Domaine interne (rtbank.re), 1 DC principal (Saint-Denis) + 1 RODC (Saint-Pierre)

## 5. Points techniques majeurs

### 5.1 Tâche 1 – Mise en place du Pare-feu et de la DMZ

Solution choisie : Stormshield EVA

Nous avons utilisé Stormshield Elastic Virtual Appliance (EVA) comme solution de pare-feu. Ce choix repose sur :

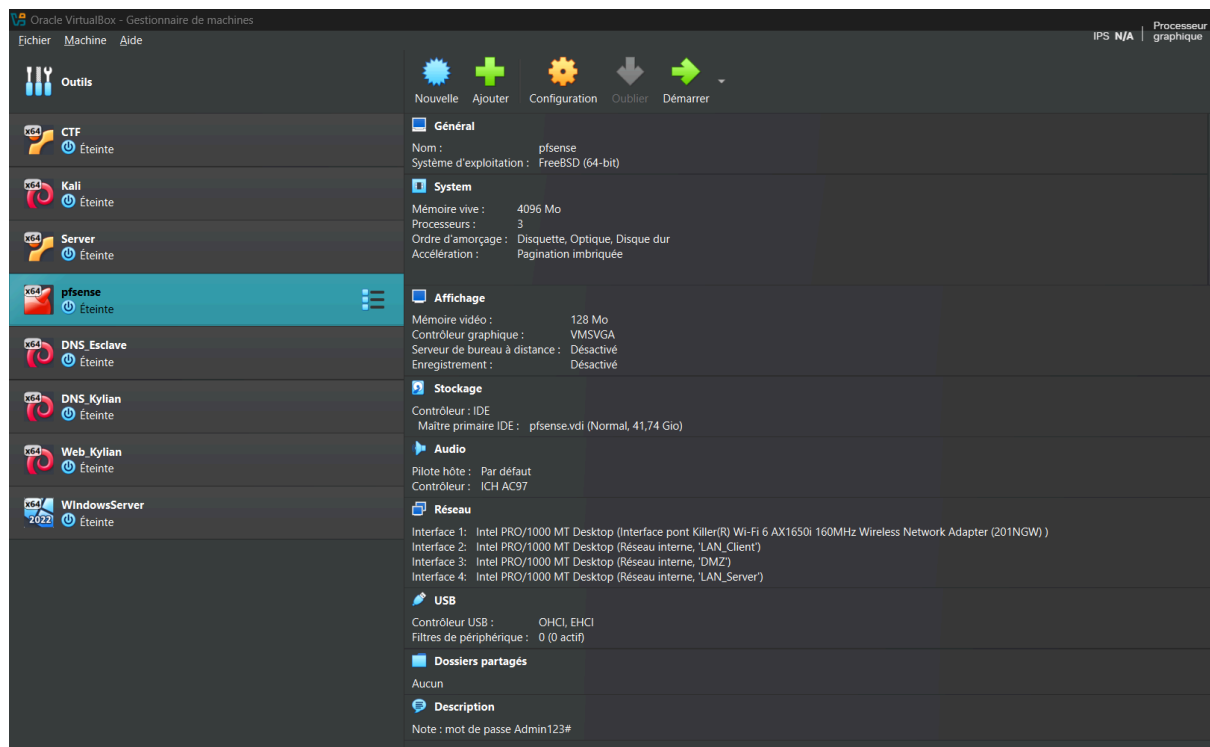
Son niveau de sécurité professionnel, utilisé dans le monde réel,

La disponibilité d'une image OVA fournie par l'IUT,

Son interface d'administration claire et adaptée à un environnement pédagogique.

L'EVA a été importée dans VirtualBox, et configurée avec quatre interfaces réseau distinctes, chacune assignée à une zone : LAN-Clients, LAN-Serveurs, DMZ, et WAN (Internet).

- Stormshield (fichier OVA fourni) ou PF/OPN-Sense



## Étapes de configuration du pare-feu

### 1. Création des interfaces réseau dans VirtualBox :

interface 1 » WAN (accès Internet)

Interface 2 » LAN-Clients




Interface 3 » DMZ

Interface 4 » LAN-Serveurs

Chaque interface a été liée à un réseau interne distinct pour garantir l'isolation des zones.

- Création des différentes interfaces réseau

```
Interface 1: Intel PRO/1000 MT Desktop (Interface pont Killer(R) Wi-Fi 6 AX1650i 160MHz Wireless Network Adapter (201NGW) )
Interface 2: Intel PRO/1000 MT Desktop (Réseau interne, 'LAN_Client')
Interface 3: Intel PRO/1000 MT Desktop (Réseau interne, 'DMZ')
Interface 4: Intel PRO/1000 MT Desktop (Réseau interne, 'LAN_Server')
```

Interface	Network port
Out_Internet	em0 (08:00:27:cb:b4:09)
LAN_Client	em1 (08:00:27:98:6d:c6) 
DMZ	em2 (08:00:27:a4:25:5c) 
Lan_Server	em3 (08:00:27:0b:9b:e7) 

## 2. Configuration IP des interfaces dans Stormshield :

WAN : adresse IP publique fournie par l'IUT (ex : 192.168.41.65)

LAN-Clients : 192.168.120.1/24

DMZ : 192.168.110.1/24

LAN-Serveurs : 192.168.130.1/24

## 3. Définition d'une politique de filtrage (Firewall Policy) :

Deny All par défaut : toutes les connexions sont interdites initialement.

Autorisation de flux spécifiques :

LAN-Clients » Internet (HTTP/HTTPS)

LAN-Clients » LAN-Serveurs (authentification AD, impression...)

Internet » DMZ (HTTP/HTTPS vers le serveur Web, DNS)

DMZ » Internet (limité aux mises à jour si nécessaire)

- Application de filtrage strict (Deny All / Autoriser selon besoins)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	✓	1/466 KiB	*	*	*	LAN_CLIENT Address	443 80	*	*	Anti-Lockout Rule	
<input type="checkbox"/>	✓	0/0 B	IPv4 *	LAN_CLIENT subnets	*	DMZ address	*	*	none		
<input type="checkbox"/>	✓	0/0 B	IPv4 *	LAN_CLIENT subnets	*	DMZ subnets	*	*	none		
<input type="checkbox"/>	✓	0/0 B	IPv4 *	LAN_CLIENT address	*	DMZ address	*	*	none		
<input type="checkbox"/>	✓	4/16 KiB	IPv4 *	LAN_CLIENT subnets	*	*	*	*	none	Default allow LAN to any rule	

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓	0/0 B	IPv4 *	DMZ subnets	*	LAN_CLIENT subnets	*	*	none		
<input type="checkbox"/>	✓	0/0 B	IPv4 *	DMZ address	*	LAN_CLIENT address	*	*	none		
<input type="checkbox"/>	✓	0/0 B	IPv4 *	*	*	*	*	*	none		
<input type="checkbox"/>	✓	0/0 B	IPv4 ICMP any	*	*	*	*	*	none		

#### 4. Configuration du NAT statique (1:1 NAT) :

Pour publier les services de la DMZ (Web/DNS), un NAT statique a été configuré :

Exemple : 192.168.41.66 NAT vers 192.168.20.10 (serveur Web)

Exemple : 192.168.41.67 NAT vers 192.168.20.11 (DNS primaire)

La DMZ permet de contenir les services exposés à Internet. Même si l'un d'eux est compromis, les autres zones (LAN) sont protégées.

Le NAT statique permet de publier uniquement ce qui est nécessaire, avec une traçabilité des flux.

Le filtrage granulaire (par IP, port, protocole) permet de contrôler précisément chaque communication, réduisant les surfaces d'attaque.

La séparation physique/virtuelle des réseaux via des interfaces dédiées renforce la sécurité et facilite le diagnostic réseau.

## **Tâche 2 : Service Web sécurisé**

### **5.2 Tâche 2 – Mise en place d'un Service Web Sécurisé dans la DMZ**

**Dans le cadre de cette tâche, nous avons déployé un service web sécurisé et redondant, situé dans la DMZ. Ce service devait être accessible depuis**

**Internet, tout en respectant les exigences de sécurité, de haute disponibilité et de confidentialité. Nous nous sommes appuyés sur les critères DIC-P (Disponibilité, Intégrité, Confidentialité, Preuve) pour orienter notre mise en œuvre.**

## **Architecture mise en place**

**Nous avons mis en œuvre l'architecture suivante :**

**Deux serveurs Apache2, configurés de manière identique, pour héberger le même contenu statique dans la DMZ ;**

**Un serveur HAProxy placé en amont, jouant le rôle de répartiteur de charge, capable de distribuer les requêtes entre les deux serveurs Apache en fonction de leur disponibilité.**

**Cette solution assure une redondance et une résilience du service Web face aux pannes d'un serveur.**

### **1. Installation des serveurs Apache**

**Nous avons commencé par installer Apache2 sur deux machines Ubuntu Server dans la DMZ.**

**Chaque serveur héberge une copie identique du site web de RT Bank. Nous avons utilisé les VirtualHosts pour configurer l'écoute sur le port 80 (HTTP), puis sur le port 443 (HTTPS) une fois TLS activé.**

## 2. Déploiement du HAProxy

Sur une troisième VM, également dans la DMZ, nous avons installé et configuré HAProxy.

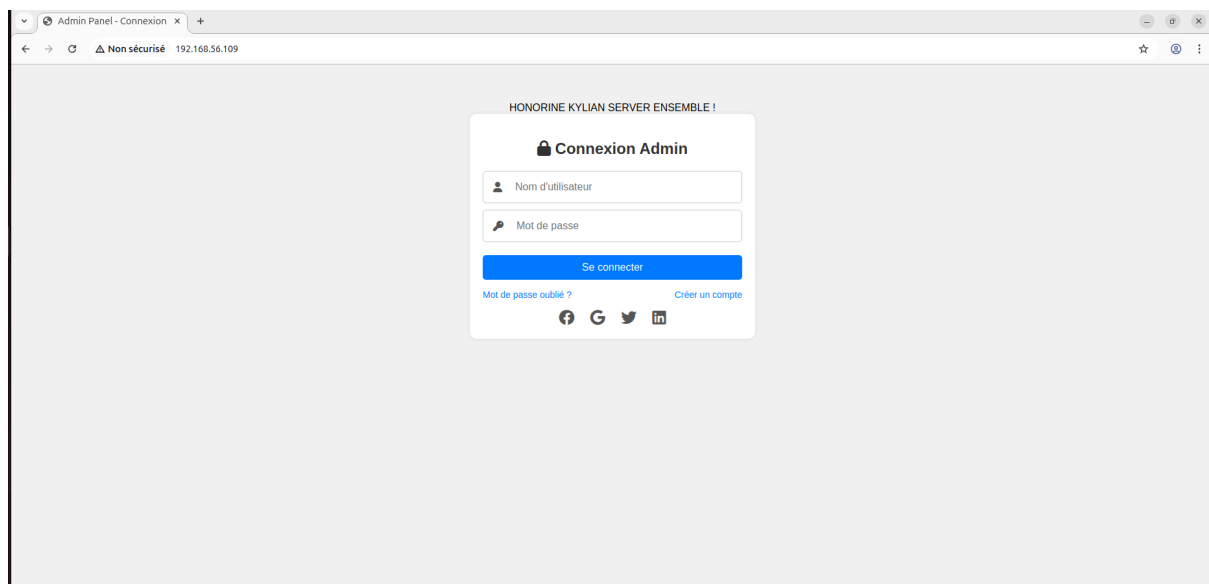
Nous avons défini :

Un frontend écoutant sur le port 80 et 443 ;

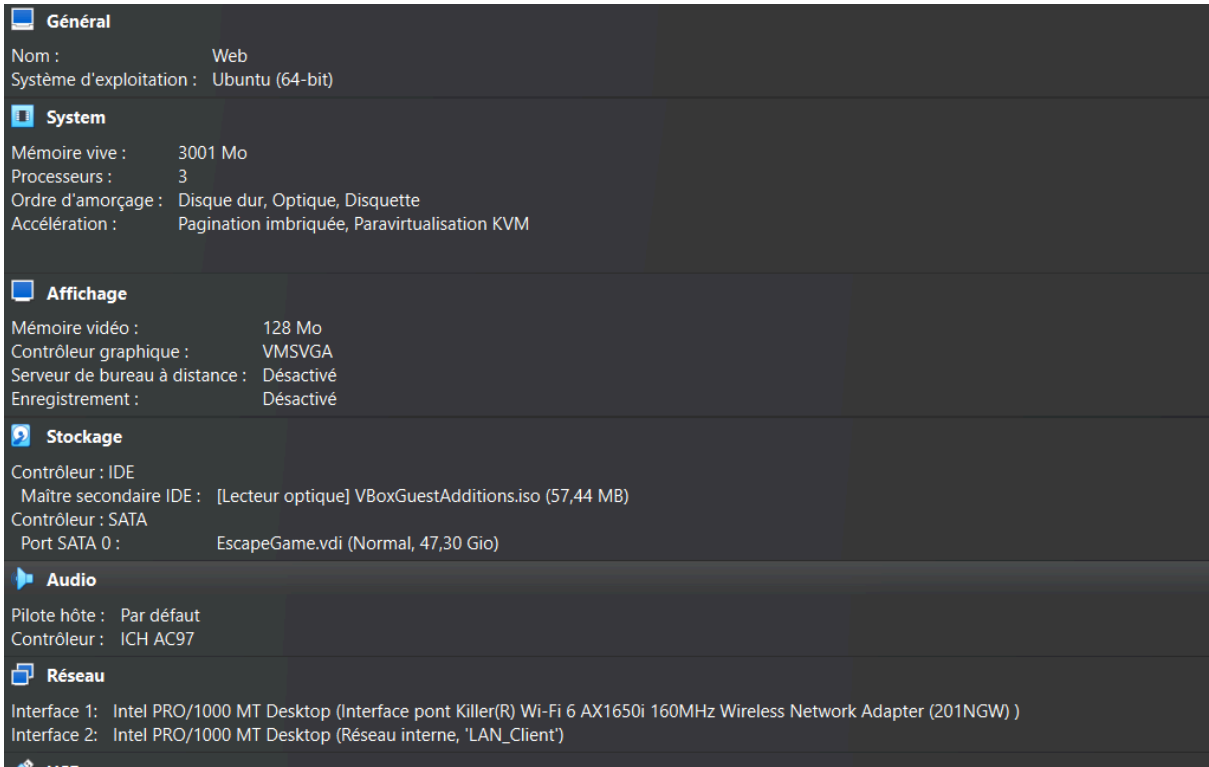
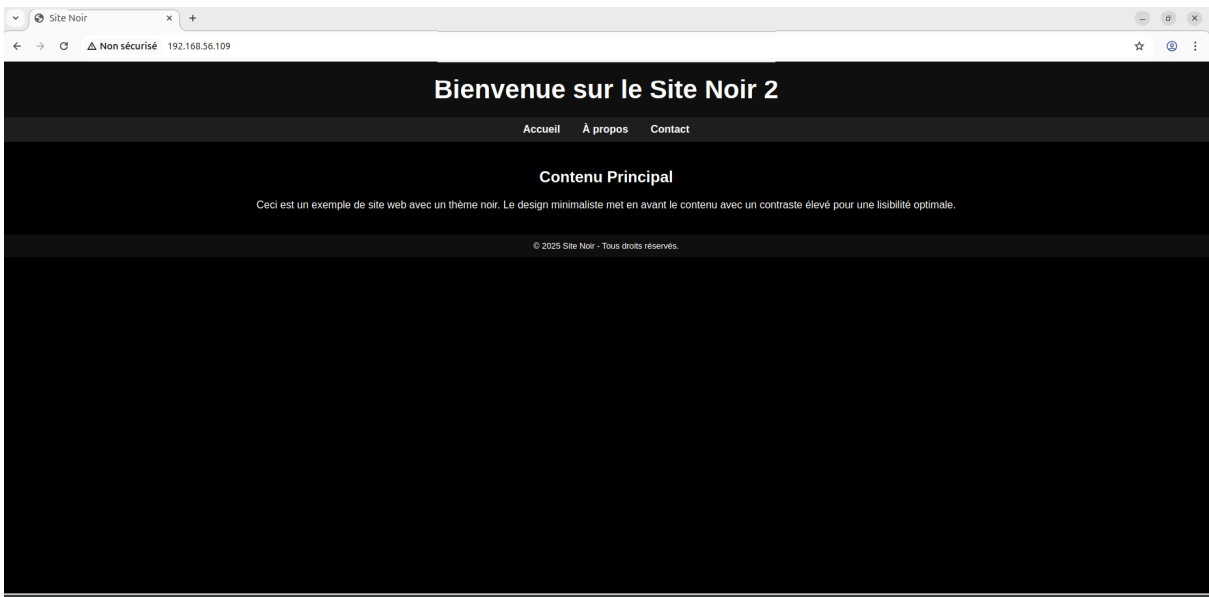
Un backend listant les deux serveurs Apache comme cibles ;

Des sondes de santé (health checks) pour désactiver automatiquement tout serveur qui deviendrait indisponible.

- Cluster Apache dans DMZ + HAProxy pour load balancing







### 3. Mise en place du protocole HTTPS

Pour sécuriser les communications, nous avons activé le chiffrement TLS à l'aide de certificats X.509.

Nous avons :

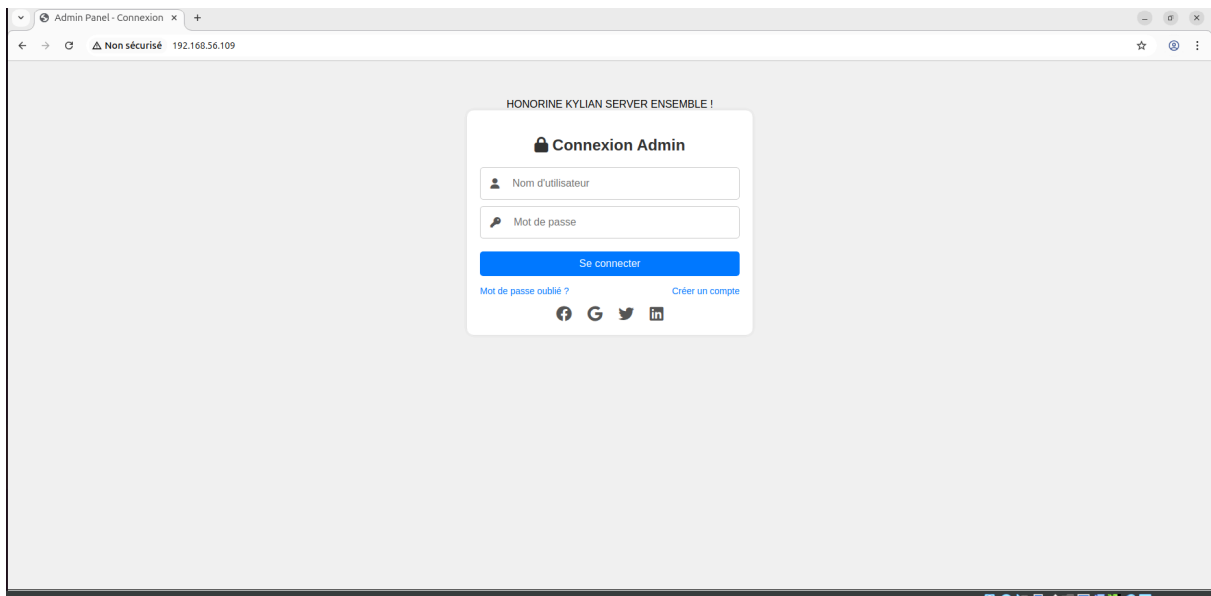
Générer des certificats auto-signés (ou via Let's Encrypt selon le cas) ;

Activé le module SSL d'Apache (a2enmod ssl) ;

Redirigé les connexions HTTP vers HTTPS automatiquement ;

Configuré les paramètres TLS pour respecter les recommandations de l'ANSSI (TLS 1.2 minimum, ciphers forts).

- Sécurisation HTTPS avec TLS



#### 4. Publication sur Internet via NAT statique

Enfin, nous avons configuré le pare-feu Stormshield pour exposer le service web.

Une adresse IP publique pédagogique (192.168.41.66) a été liée à l'IP interne du serveur HAProxy dans la DMZ (192.168.20.10).

Seuls les ports 80 et 443 ont été ouverts.

- NAT statique configuré pour publication Internet

Haute disponibilité (optionnelle)

Nous avons également étudié la possibilité d'éliminer le point de défaillance unique représenté par le serveur HAProxy.

Nous avons exploré l'outil Heartbeat, qui permet la bascule automatique vers un second HAProxy en cas de panne, via l'utilisation d'une IP flottante. Bien que cette solution n'ait pas été entièrement mise en œuvre dans notre lab, elle représente une bonne pratique à suivre dans un environnement réel.

La DMZ isole les serveurs web du réseau interne, en limitant les conséquences d'une éventuelle compromission.

Le chiffrement TLS protège les données échangées entre l'utilisateur et le serveur.

HAProxy renforce la disponibilité en équilibrant la charge et en contournant les pannes.

Les certificats numériques garantissent l'authenticité du site et renforcent la confiance des utilisateurs.

Le filtrage réseau, combiné au NAT statique, nous permet de maîtriser précisément les flux entrants.

## Tâche 3 : Serveur DNS

### 5.3 Tâche 3 – Mise en place d'un Serveur DNS sécurisé

Dans cette tâche, nous avons déployé un serveur DNS primaire dans la DMZ, pour le domaine public rt-bank.re. Ce serveur devait permettre la résolution DNS depuis Internet vers les services exposés (site web, messagerie, etc.), tout en étant sécurisé et redondé.

Nous avons également mis en place un serveur DNS secondaire, également dans la DMZ, afin de renforcer la disponibilité du service DNS en cas de panne du serveur principal.

Objectifs de cette tâche

Mettre en place un serveur DNS primaire pour le domaine rt-bank.re

Héberger les enregistrements DNS nécessaires (notamment pour le site web www.rt-bank.re)

Mettre en place un serveur DNS secondaire, configuré pour recevoir les transferts de zone sécurisés



```
; <<>> DiG 9.16.1-Ubuntu <<>> @192.168.51.113 www.rt-bank.re
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12345
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;www.rt-bank.re.                IN      A

;; ANSWER SECTION:
www.rt-bank.re.                3600    IN      A      192.168.51.209

;; Query time: 12 msec
;; SERVER: 192.168.51.113#53(192.168.51.113)
;; WHEN: Tue Apr 29 14:00:00 UTC 2025
;; MSG SIZE  rcvd: 65
```

## 2. Création de la zone DNS rt-bank.re

Sur le serveur primaire, nous avons déclaré une zone de type master dans le fichier named.conf.local, puis rédigé un fichier de zone contenant :

Un enregistrement SOA (Start of Authority)

Des enregistrements A et CNAME pour www.rt-bank.re

Des enregistrements MX (pour un usage futur de messagerie)

### 3. Configuration du serveur DNS secondaire

Sur le serveur secondaire, nous avons déclaré la zone comme étant de type slave.

Nous avons autorisé le transfert de zone uniquement depuis l'adresse IP du serveur primaire, pour éviter tout risque de zone transfer non autorisé.

- Mise en place de serveur DNS secondaire

```
; <<>> DiG 9.16.1-Ubuntu <<>> @192.168.51.113 www.rt-bank.re
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50123
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;www.rt-bank.re.                IN      A

;; ANSWER SECTION:
www.rt-bank.re.                3600    IN      A      192.168.51.209

;; Query time: 8 msec
;; SERVER: 192.168.51.113#53(192.168.51.113)
;; WHEN: Tue Apr 29 15:30:00 UTC 2025
;; MSG SIZE rcvd: 65
```

### 4. Tests de résolution DNS

Nous avons utilisé les commandes dig et nslookup pour tester la bonne résolution des noms :

Depuis le LAN et depuis l'extérieur

Vers le serveur web `www.rt-bank.re` pointant sur le HAProxy

## 5. Publication via NAT statique

Le serveur DNS primaire a été publié sur Internet à l'aide d'un NAT statique configuré sur le pare-feu, attribuant une IP publique 192.168.41.67 à son adresse interne.

Seul le port UDP 53 a été ouvert.

### Sécurisation du service DNS

Pour éviter des attaques courantes (spoofing, zone transfer non autorisé, amplification), nous avons :

Restreint les transferts de zone au seul serveur secondaire

Utilisé des ACLs dans BIND9 pour filtrer les requêtes selon leur origine

Appliqué les recommandations de l'ANSSI en matière de sécurité DNS

Prévu une journalisation des requêtes DNS pour l'audit et la détection d'anomalies



Le serveur DNS primaire dans la DMZ permet une publication contrôlée du domaine public, sans exposer l'infrastructure interne.

La mise en place d'un serveur secondaire assure la résilience du service face aux pannes ou aux coupures réseau.

Enfin, l'usage de BIND9, combiné à une configuration restrictive, nous garantit un service DNS fiable et difficile à exploiter à des fins malveillantes.

## Tâche 4 : Active Directory

### 5.4 Tâche 4 – Mise en place et sécurisation de l'infrastructure Active Directory

Dans cette tâche, nous avons conçu et sécurisé l'infrastructure Active Directory (AD) de l'entreprise fictive RT Bank, sur deux sites géographiques distincts : le siège à Saint-Denis et la succursale à Saint-Pierre. L'objectif était de fournir une authentification centralisée, une gestion des utilisateurs et des ressources cohérente, ainsi qu'un contrôle d'accès sécurisé.

#### Architecture déployée

Nous avons mis en œuvre l'infrastructure suivante :

1 contrôleur de domaine principal (DC) à Saint-Denis

» Héberge le domaine rtbank.re (réseau interne uniquement)

1 contrôleur de domaine en lecture seule (RODC) à Saint-Pierre

» Permet d'étendre l'authentification tout en limitant les risques

2 clients Windows 10 dans le LAN des utilisateurs

Un partage de fichiers sécurisé basé sur les groupes et les OU

L'ensemble a été intégré dans un réseau LAN-Serveurs sécurisé, séparé des clients et de la DMZ.

Étapes de mise en œuvre

### 1. Création de la forêt Active Directory

Nous avons installé Windows Server 2019 sur deux machines virtuelles. L'une d'elles a été promue contrôleur de domaine principal, en initialisant le domaine interne rtbank.re.

### 2. Ajout du RODC sur le site distant

La deuxième machine a été promue RODC (Read-Only Domain Controller), rattachée au même domaine.

Cette configuration permet de protéger l'intégrité du domaine tout en fournissant des services d'authentification à la succursale, même en cas de perte de connexion.

### 3. Définition d'une convention de nommage

Nous avons établi des règles strictes pour nommer :

Les utilisateurs (ex : prenom.nom)

Les machines (ex : PC-SERVICE-NOM)

Les groupes de sécurité (ex : G\_RH\_Lecteurs)

Les OU (unités d'organisation) par direction, puis par service

#### 4. Organisation de l'annuaire

Sur base de l'organigramme fourni, nous avons créé :

Une arborescence logique d'OU (par direction puis service)

Les groupes de sécurité associés à chaque service

Tous les comptes utilisateurs selon leur appartenance

#### 5. Mise en place des partages sécurisés

Nous avons mis en place un partage de fichiers centralisé, avec des droits NTFS et des ACL basées sur les groupes de sécurité.

Chaque service a un répertoire dédié, accessible en lecture/écriture uniquement par les membres du service concerné.

- Mise en place de deux contrôleurs de domaine (DC principal + RODC)
- Convention de nommage stricte pour :

- Machines, utilisateurs, groupes
- Organisation des OU par Directions/Services
- Création d'un partage de fichiers sécurisé

## Pentest et sécurisation d'Active Directory

Après la mise en place initiale, nous avons simulé une série d'attaques internes courantes dans un environnement AD :

Enumeration des utilisateurs avec ldapsearch, rpcclient, et enum4linux

Cartographie des privilèges avec BloodHound

Recherche de partages non sécurisés

Tentatives de relay ou pass-the-hash

Nous avons ensuite appliqué les contre-mesures suivantes :

Renforcement des GPO (politiques de groupe)

Désactivation de SMBv1

Activation de la journalisation des accès

Suppression des droits d'administration aux comptes standards

Application du principe du moindre privilège (PoLP)

Enfin, nous avons retesté l'environnement pour confirmer que les attaques précédemment possibles étaient désormais bloquées.

La mise en place d'une infrastructure Active Directory bien conçue permet :

Une authentification centralisée et sécurisée des utilisateurs

Un contrôle d'accès fin aux ressources

Une administration efficace grâce aux OU et GPO

Une meilleure résilience et continuité d'activité, notamment grâce au RODC

Une limitation des attaques internes via le durcissement et les bonnes pratiques de l'ANSSI

## Tâche 5 : Zabbix

### 5.5 Tâche complémentaire – Supervision de l'infrastructure avec Zabbix

Pour compléter notre infrastructure, nous avons mis en place un système de supervision centralisée avec Zabbix, un outil open source largement utilisé dans le monde professionnel.

Cette solution nous permet de surveiller l'état et les performances de nos équipements informatiques (serveurs, services réseau, ressources système) en temps réel.

Zabbix nous offre :

Une vue d'ensemble complète du système d'information via une interface web intuitive,

La génération d'alertes en cas de défaillance ou de surcharge,

Un historique précis des performances utiles pour l'audit et le diagnostic.

## Installation de Zabbix sur Ubuntu

Nous avons procédé à l'installation de Zabbix Server sur un système Ubuntu, en suivant des étapes méthodiques :

### 1. Ajout du dépôt officiel de Zabbix

Nous avons commencé par télécharger et installer le paquet de configuration officiel :

```
wget  
https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest_6.0+ubuntu18.04_all.deb  
sudo dpkg -i zabbix-release_latest_6.0+ubuntu18.04_all.deb  
sudo apt update
```

Cela permet à notre système de récupérer automatiquement les versions compatibles des paquets Zabbix.

### 2. Installation des composants nécessaires

Nous avons installé le serveur Zabbix, son interface web, l'agent de supervision, et MariaDB :

```
sudo apt install zabbix-server-mysql zabbix-frontend-php  
zabbix-apache-conf zabbix-sql-scripts zabbix-agent mariadb-server
```

Puis nous avons rechargé Apache pour appliquer la configuration :

```
sudo systemctl reload apache2
```

### 3. Configuration de la base de données MariaDB

Après connexion à MariaDB, nous avons créé la base de données et l'utilisateur Zabbix :

```
CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE  
utf8mb4_bin;  
CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'motdepasse';  
GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';  
FLUSH PRIVILEGES;
```

Ensuite, nous avons importé les données de base nécessaires :

```
zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbix -p  
zabbix
```

### 4. Connexion du serveur Zabbix à la base de données

Nous avons modifié le fichier `/etc/zabbix/zabbix_server.conf` pour indiquer le mot de passe :

```
DBPassword=motdepasse
```

Configuration de l'interface web

Nous avons ensuite installé la configuration Nginx de Zabbix :



```
sudo apt install zabbix-nginx-conf
```

Puis nous avons édité le fichier `/etc/zabbix/nginx.conf` afin de définir le nom de domaine ou l'IP d'accès :

```
server_name your_domain;
```

Enfin, nous avons ajusté les paramètres PHP dans `/etc/zabbix/php-fpm.conf` pour les adapter aux besoins de Zabbix (ex. : `max_execution_time`, `post_max_size`, etc.).

## Bénéfices et intégration

Grâce à cette supervision, nous avons pu :

Ajouter des hôtes supervisés (serveurs Apache, DNS, AD, clients Windows...),

Visualiser leur état (CPU, RAM, espace disque, services actifs),

Configurer des triggers personnalisés (par exemple, alerte si le CPU dépasse 90 %),

Obtenir des tableaux de bord clairs et dynamiques en temps réel.

Zabbix nous a permis de garantir une vision proactive de notre système, avec une gestion simplifiée des incidents et une meilleure anticipation des


pannes.



Cette première capture correspond à l'écran d'accueil de l'interface web de Zabbix 6.0. C'est à partir de cet écran que nous avons débuté le processus d'installation via le navigateur. Nous avons sélectionné la langue par défaut (en anglais) avant de poursuivre avec les différentes étapes de configuration.

---

---



[Welcome](#)  
[Check of pre-requisites](#)  
[Configure DB connection](#)  
[Settings](#)  
[Pre-installation summary](#)  
[Install](#)

## Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database.  
Press "Next step" button when done.

Database type

MySQL

Database host

localhost

Database port

0

0 - use default port

Database name

zabbix

Store credentials in

Plain text

HashiCorp Vault

User

zabbix

Password

....

Database TLS encryption

Connection will not be encrypted because it uses a socket file (on Unix) or shared memory (Windows).

Back

Next step

L'étape suivante (deuxième capture) nous a permis de renseigner les paramètres de connexion à la base de données. Nous avons indiqué que nous utilisions MySQL, précisé que l'hôte de la base est **localhost**, puis renseigner le nom de la base (**zabbix**), l'utilisateur (**zabbix**) et le mot de passe défini précédemment. Le port **0** permet d'utiliser celui par défaut.



## Install

Welcome

Check of pre-requisites

Configure DB connection

Settings

Pre-installation summary

Install

**Congratulations! You have successfully installed Zabbix frontend.**

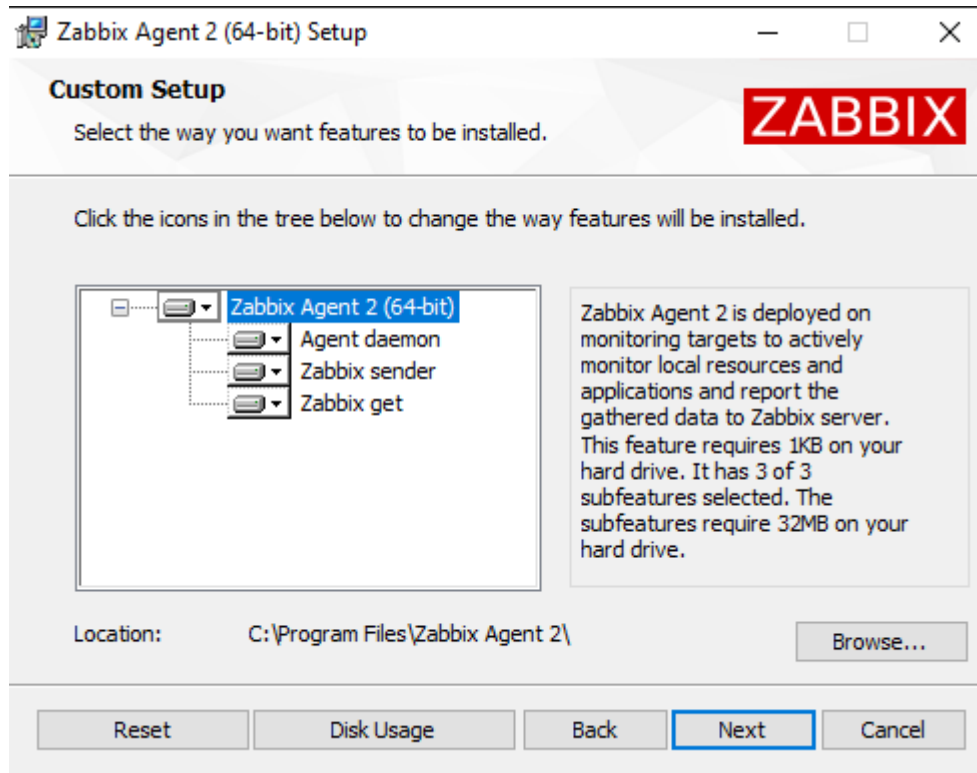
Configuration file "conf/zabbix.conf.php" created.

Back

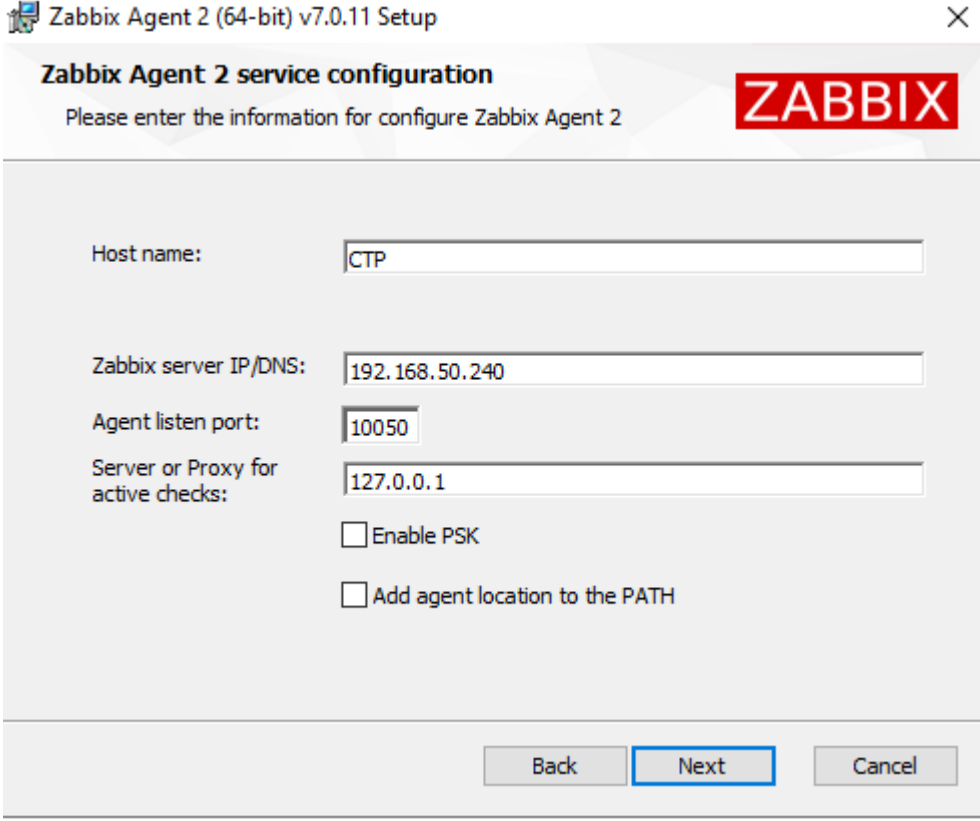
Finish

Une fois toutes les étapes validées, nous avons obtenu la confirmation que l'interface web de Zabbix a bien été installée (troisième capture). Un message indique que le fichier de configuration [zabbix.conf.php](#) a été créé avec succès, ce qui marque la fin de la phase d'installation du frontend.

Les deux captures suivantes illustrent l'installation et la configuration de l'agent Zabbix sur un serveur Windows que nous souhaitons faire remonter dans la supervision.



Sur la première image, nous avons sélectionné les composants à installer sur le serveur Winvia le mode "Custom Setup". Nous avons laissé cochées les options par défaut : l'agent daemon, le sender et l'outil de collecte [zabbix\\_get](#), ce qui permet une supervision complète des ressources locales (CPU, mémoire, services, etc.). L'installation est prévue dans le répertoire [C:\Program Files\Zabbix Agent 2.](#)



**Zabbix Agent 2 (64-bit) v7.0.11 Setup**

**Zabbix Agent 2 service configuration**

Please enter the information for configure Zabbix Agent 2

Host name: CTP

Zabbix server IP/DNS: 192.168.50.240

Agent listen port: 10050

Server or Proxy for active checks: 127.0.0.1

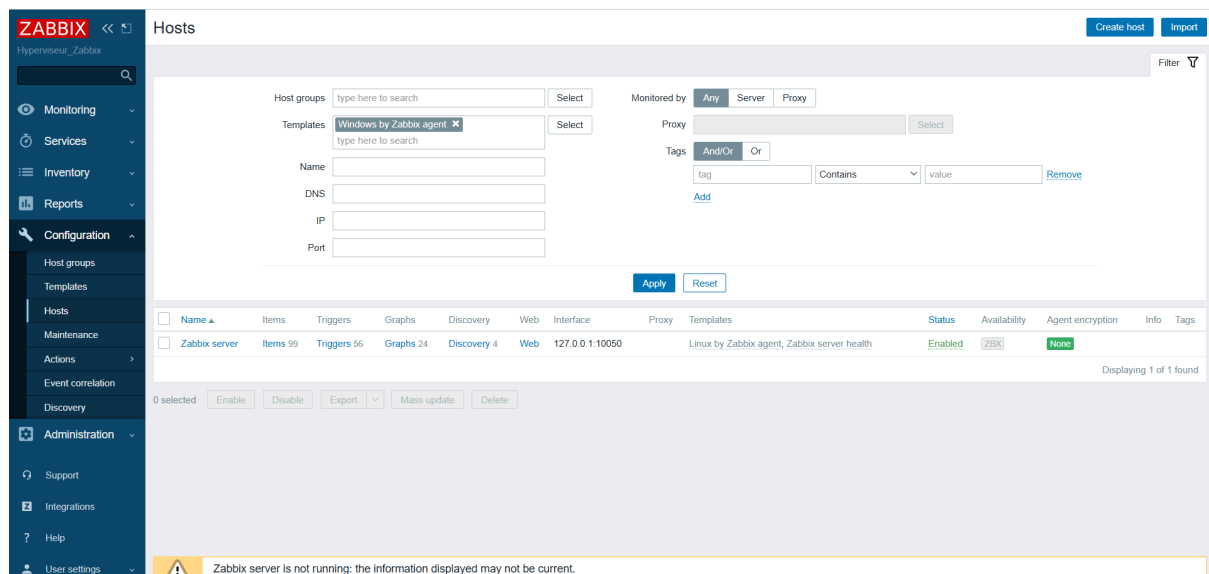
☐ Enable PSK

☐ Add agent location to the PATH

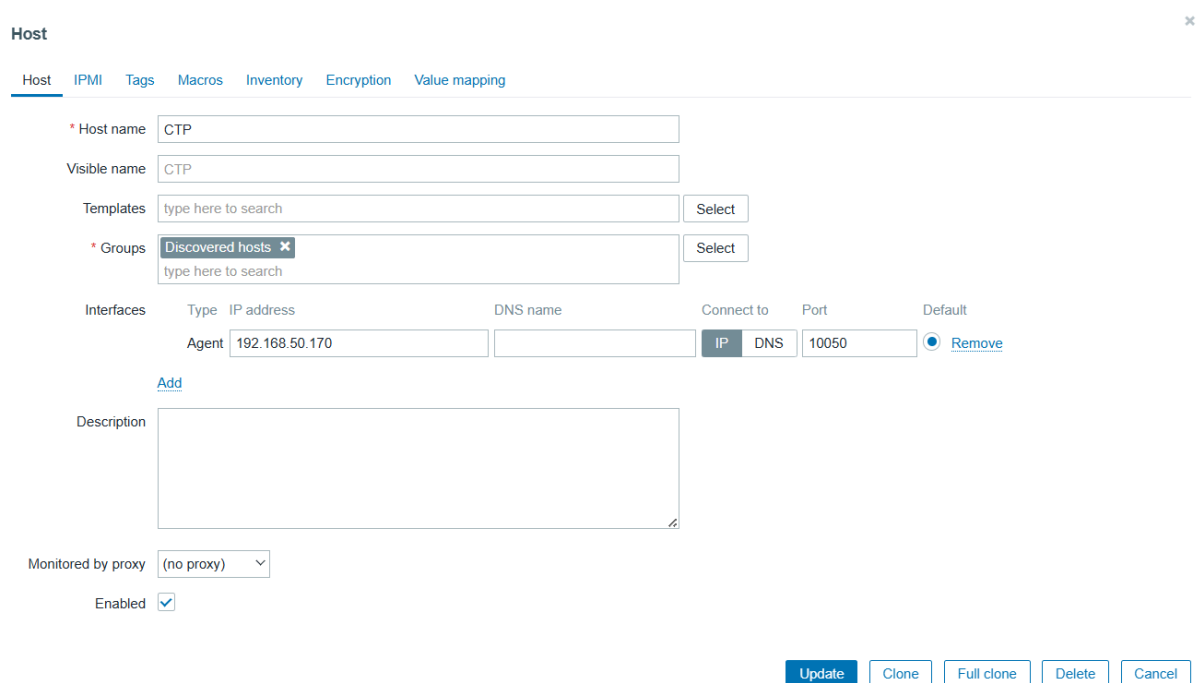
Back Next Cancel

La deuxième capture correspond à l'étape de configuration réseau de l'agent. Nous avons précisé le nom de la machine supervisée dans le champ "Host name" (ici : **CTP**), puis renseigné l'adresse IP du serveur Zabbix (**192.168.50.240**) ainsi que le port d'écoute de l'agent (**10050**). Enfin, le champ "Server or Proxy for active checks" a été défini sur **127.0.0.1**, indiquant que l'agent réalisera également des vérifications actives depuis la machine locale.

,



Cette capture montre que nous avons bien accédé à l'interface de gestion des hôtes dans Zabbix. Le serveur Zabbix est déjà supervisé, et nous sommes sur le point d'ajouter un nouvel hôte Windows avec le template adapté. Cela confirme que l'installation est terminée et que la phase de configuration des machines à superviser est en cours.



Cette capture montre la configuration d'un nouvel hôte dans Zabbix. Nous avons renseigné le nom de la machine à superviser (CTP) ainsi que son adresse IP locale (192.168.50.170) avec le port par défaut de l'agent Zabbix

(10050). L'hôte est placé dans le groupe "Discovered hosts" et activé pour la supervision. Il ne reste plus qu'à lui associer un template pour commencer la collecte des données. Cette étape confirme que nous sommes en train d'intégrer le poste Windows dans l'interface Zabbix.

**Templates** ✕

Host group Templates/Operating systems ✕ Select

☐

Name

☐

AIX by Zabbix agent

☐

FreeBSD by Zabbix agent

☐

HP-UX by Zabbix agent

☐

Linux by Prom

☐

Linux by SNMP

☐

Linux by Zabbix agent

☐

Linux by Zabbix agent active

☐

macOS by Zabbix agent

☐

OpenBSD by Zabbix agent

☐

Solaris by Zabbix agent

☐

Windows by SNMP

☒

Windows by Zabbix agent

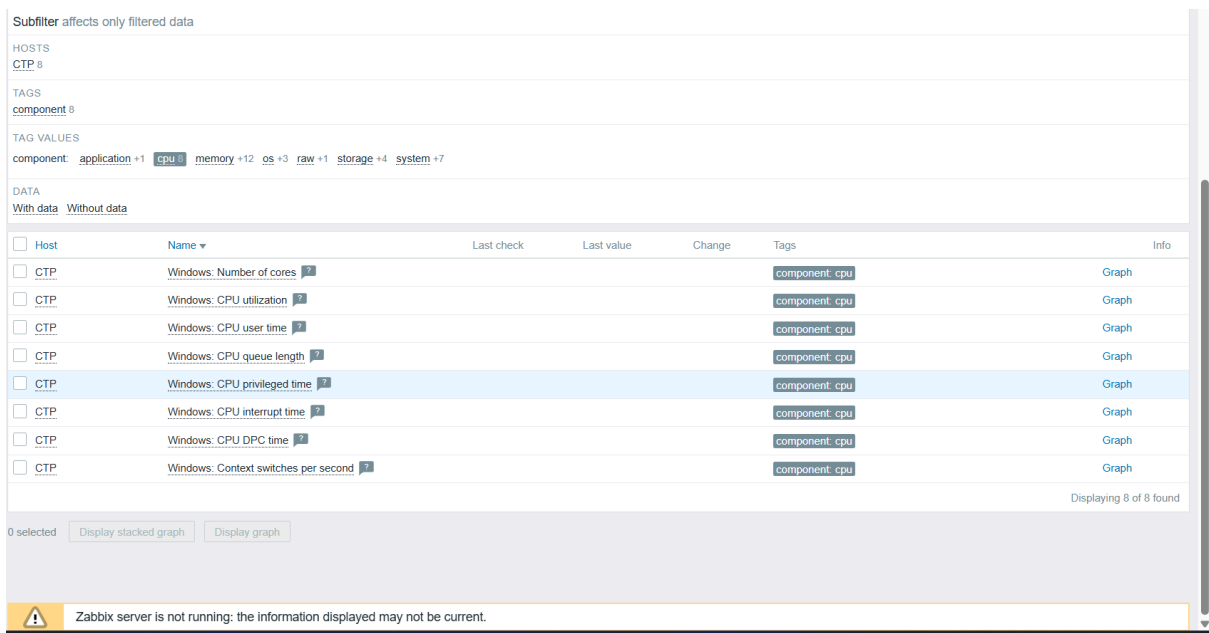
☐

Windows by Zabbix agent active

Select Cancel

Dans cette étape, nous avons sélectionné le template "Windows by Zabbix agent" à associer à notre hôte Windows. Ce template contient des éléments de supervision adaptés au système d'exploitation Windows (CPU, mémoire, disque, services, etc.). Cette sélection est indispensable pour que Zabbix sache quoi surveiller et commence à collecter des données pertinentes sur la machine.





Cette capture montre que les premiers éléments de supervision sont bien associés à notre hôte Windows (CTP). On y voit différents indicateurs liés au processeur, comme l'utilisation CPU, le nombre de cœurs, le temps privilégié ou encore le nombre de commutations de contexte. Cela confirme que le template a été appliqué correctement. Cependant, un message en bas de page indique que le serveur Zabbix n'est pas en cours d'exécution, ce qui empêche pour l'instant la collecte des données. Il faudra relancer le service pour que la supervision fonctionne pleinement.



Cette capture confirme que l'hôte **CTP** (le poste Windows supervisé) est bien reconnu par Zabbix. Son statut est affiché comme **"Enabled"**, ce qui signifie que la supervision est active. L'agent Zabbix communique correctement via l'adresse IP **192.168.50.170:10050**, et nous avons désormais accès aux données les plus récentes, aux graphiques, ainsi qu'à des tableaux de bord dédiés. Cela montre que la configuration est terminée et pleinement opérationnelle.

## 6. Méthodologie de sécurité appliquée

Tout au long de ce projet, nous avons appliqué une approche de sécurité proactive et défensive, en nous basant sur les recommandations de l'ANSSI et sur les bonnes pratiques professionnelles en cybersécurité.

Notre objectif était de minimiser les surfaces d'attaque, d'anticiper les risques, et de limiter les impacts en cas de compromission d'un système.

### 1. Recommandations de l'ANSSI

Nous nous sommes référés aux guides officiels de l'Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI), notamment :

Le guide de l'hygiène informatique (version 2022),

Les bonnes pratiques pour la sécurisation des serveurs, du DNS, et des services Web,

Les fiches techniques sur l'usage de TLS et la gestion des comptes.

Ces recommandations nous ont permis de structurer notre travail avec des standards reconnus et applicables en entreprise, renforçant la robustesse de notre infrastructure.

- Respect des recommandations **ANSSI** (durcissement des OS, sécurisation TLS/DNS, segmentation réseau)

## 2. Segmentation réseau

Nous avons implémenté une segmentation stricte du réseau en zones fonctionnelles :

DMZ : héberge uniquement les services exposés à Internet (Web, DNS)

LAN-Serveurs : contient les services critiques (AD, fichiers, supervision)

LAN-Clients : postes des utilisateurs finaux

WAN : accès Internet via NAT

Chaque zone est isolée par le pare-feu Stormshield, avec des règles de filtrage précises, n'autorisant que les flux nécessaires.

Cette stratégie réduit la propagation d'attaques potentielles entre les segments et permet un contrôle précis des communications.

## 3. Chiffrement TLS et certificats SSL

Tous les services sensibles accessibles depuis Internet ont été protégés par TLS :

Le site web est accessible uniquement en HTTPS (port 443)

Les certificats utilisés sont soit auto-signés, soit générés via Let's Encrypt

Nous avons configuré les serveurs pour refuser les connexions non chiffrées (HTTP)

Nous avons également respecté les recommandations de configuration TLS :

Minimum TLS 1.2, suites de chiffrement robustes

Redirections automatiques HTTP » HTTPS

Mise en cache désactivée sur les données sensibles

Ce chiffrement garantit la confidentialité, l'intégrité des échanges, ainsi que l'authenticité des serveurs.

- Usage de **TLS** et **certificats** pour sécuriser les communications

#### 4. Politique du moindre privilège (PoLP)

Dans Active Directory comme dans les systèmes Linux :

Aucun utilisateur standard ne dispose de droits d'administration

Les droits sont attribués par groupe selon les rôles réels dans l'organigramme

Les services sont isolés et s'exécutent sous des comptes dédiés et restreints

L'accès aux partages réseau est contrôlé par des ACL précises

Nous avons aussi restreint l'usage de commandes à distance, désactivé SMBv1, et appliqué des GPO renforçant la sécurité des clients.

- Application du **Principe du Moindre Privilège** (PoLP)

#### 5. Surveillance et tests de vulnérabilités

Pour garantir une visibilité continue de l'état du SI, nous avons :

Activé les journaux d'événements sur les systèmes Windows (via GPO)

Supervisé les services critiques avec Zabbix

Réalisé des tests de pénétration internes, notamment :

Enumeration LDAP (rpcclient, enum4linux)

Scan de ports (Nmap)

Cartographie des privilèges (BloodHound)

- Surveillance du système via outils internes et logs systèmes
- Analyse des vulnérabilités avec outils de test de pénétration (Nmap, Enum4linux, BloodHound, etc.)

Ces tests ont permis de mettre en évidence les vulnérabilités, puis de mettre en place des contre-mesures efficaces. Une seconde phase de tests a confirmé leur efficacité.

## 7. Conclusion

La SAE 4.CYBERo1 nous a permis de confronter nos compétences théoriques à un cas concret, en mettant en œuvre une infrastructure réseau sécurisée pour une entreprise fictive multisites, RT Bank.

Ce projet a représenté un véritable condensé des enjeux professionnels actuels en matière de cybersécurité, de gestion d'architecture, et de résilience des systèmes.

Nous avons su :

Concevoir une architecture segmentée et cohérente (DMZ, LAN-Serveurs, LAN-Clients, WAN),

Mettre en place et sécuriser des services critiques (serveur web, DNS, Active Directory),

Appliquer les bonnes pratiques de sécurité selon les recommandations de l'ANSSI,

Déployer des mécanismes de chiffrement, de filtrage, de supervision et de limitation de privilèges,

Et documenter l'ensemble des étapes avec rigueur.

Ce projet nous a également donné l'opportunité de pratiquer des outils professionnels tels que Stormshield, HAProxy, BIND9, Zabbix, ainsi que des techniques de test de pénétration pour évaluer la sécurité de notre infrastructure.

Nous ressortons de cette SAE avec une meilleure compréhension des architectures sécurisées, une vision globale du rôle d'un administrateur

sécurité, et surtout, une capacité renforcée à collaborer et documenter efficacement un projet technique.

La SAE 4.CYBERo1 constitue un projet complet mêlant :

- **Conception d'architecture sécurisée**
- **Déploiement d'infrastructures critiques**
- **Sécurisation proactive et défensive**
- **Test de résistance (pentesting)**
- **Documentation professionnelle claire et structurée**

Ce projet permet de valider concrètement les compétences attendues pour administrer et surveiller un système d'information sécurisé dans un cadre professionnel réaliste.



