SECURISATION SERVEUR LINUX — MEDIASCHOOL

Institution : Mediaschool Nice — IRIS

---

1. CONTEXTE ET OBJECTIF

Dans le cadre du BTS SIO option SISR, ce projet consiste à déployer et sécuriser un serveur Linux destiné à l'environnement pédagogique de l'école. L'objectif est de construire une infrastructure serveur stable, durcissée et documentée, servant de base propre avant mise en production sur le vrai serveur de l'école.

La solution fournie une infrastructure orientée sécurité et gestion des identités, intégrant :

- Un VPN sécurisé (WireGuard) pour les accès distants d'administration
- Un annuaire LDAP (OpenLDAP) pour la gestion centralisée des identités
- Un antivirus en conteneur (ClamAV) pour la protection des fichiers
- Un pare-feu durci (UFW + nftables) pour le contrôle des accès réseau

L'ensemble est déployé via Vagrant sur une VM Ubuntu 22.04, avec les services orchestrés par Docker Compose.

Durée de réalisation : 23 jours

2. ARCHITECTURE GLOBALE

La solution est entièrement conteneurisée et déployée sur une unique machine virtuelle Ubuntu provisionnée automatiquement par Vagrant.

VM Rôle Configuration

SRV-Mediaschool Serveur sécurisé Ubuntu 22.04 IP : 192.168.56.10 / RAM : 2 Go / 2 vCPU

Services déployés via Docker Compose :

Service Port(s) Rôle

WireGuard 51820/UDP VPN sécurisé pour accès distants

wg-easy 51821/TCP Interface web de gestion WireGuard

OpenLDAP 389, 636 Annuaire LDAP — gestion des identités

phpLDAPadmin 8080/TCP Interface web d'administration LDAP

ClamAV 3310/TCP Service antivirus temps réel

Arborescence du projet :

```
SRV_Mediaschool_IRIS/
 ├── Vagrantfile
 ├── docker-compose.yml
 ├── .env
 ├── ldap/
|   ├── config/
|   ├── database/
|   └── certs/
```

```
├── wireguard/
|   └── data/
└── clamav/
    └── a_scanner/
```

3. TECHNOLOGIES UTILISEES

Technologie	Version / Détail
Ubuntu	22.04 LTS (jammy)
Vagrant	Provisionnement automatisé de la VM
VirtualBox	Hyperviseur de type 2
Docker	Moteur de conteneurisation
Docker Compose	Orchestration des services en conteneurs
WireGuard	VPN moderne — protocole UDP
wg-easy	Interface web de gestion des clients WireGuard
OpenLDAP	Annuaire LDAP — gestion des utilisateurs
phpLDAPadmin	Interface web pour OpenLDAP
ClamAV	Antivirus open source
UFW	Pare-feu (Uncomplicated Firewall)
nftables	Couche de filtrage bas niveau

4. DESCRIPTION DES COMPOSANTS

4.1 WireGuard VPN (wg-easy)

WireGuard est un VPN moderne basé sur le protocole UDP, reconnu pour sa simplicité et ses performances. Il est déployé via l'image Docker wg-easy qui intègre une interface web de gestion permettant de créer et gérer les profils clients VPN.

Le VPN permet aux administrateurs d'accéder à distance aux interfaces web internes (phpLDAPadmin, etc.) de manière sécurisée sans exposer ces services sur Internet.

Accès à l'interface wg-easy : http://192.168.56.10:51821

Authentification : mot de passe défini dans le fichier .env

4.2 OpenLDAP

OpenLDAP fournit un service d'annuaire LDAP pour la gestion centralisée des identités utilisateurs. Le domaine configuré est ldap.local. L'annuaire peut être utilisé pour l'authentification centralisée des services de l'école.

Accès phpLDAPadmin : http://192.168.56.10:8080

Identifiant administrateur : cn=admin,dc=ldap,dc=local

4.3 ClamAV

ClamAV est déployé en conteneur Docker et fournit un service d'analyse antivirus accessible via le port 3310. Il peut scanner les fichiers déposés sur le serveur ou recevoir des requêtes de scan depuis d'autres services.

4.4 Pare-feu UFW

UFW est configuré automatiquement pour autoriser uniquement les ports nécessaires aux services déployés. Tous les ports non listés sont rejetés par défaut.

Règles UFW configurées :

- 22/TCP : SSH
- 389/TCP : LDAP
- 636/TCP : LDAPS (LDAP sécurisé)
- 8080/TCP : phpLDAPadmin
- 51821/TCP : Interface WireGuard
- 51820/UDP : WireGuard VPN

5. FONCTIONNEMENT TECHNIQUE

5.1 Provisionnement automatique (Vagrant)

Le fichier Vagrantfile définit :

- L'image de base Ubuntu 22.04 LTS
- La configuration réseau (interface Host-Only 192.168.56.10)
- Les ressources allouées (2 Go RAM, 2 vCPU)
- Les scripts de provisionnement automatique (installation Docker, Docker Compose, démarrage des services)

La commande vagrant up crée et configure la VM automatiquement depuis un état vierge.

5.2 Orchestration Docker Compose

Le fichier docker-compose.yml définit l'ensemble des services, leurs images, les ports exposés, les volumes de persistance et les variables d'environnement (référencées depuis le fichier .env).

5.3 Variables d'environnement

Un fichier .env centralize les configurations sensibles :

- Mot de passe de l'interface WireGuard
- Mot de passe administrateur OpenLDAP
- Paramètres de domaine LDAP

Ce fichier est exclu du versionnement (listé dans .gitignore).

5.4 Gestion des utilisateurs LDAP

Les comptes utilisateurs sont créés et gérés via l'interface phpLDAPadmin. Chaque utilisateur est un objet LDAP dans l'arborescence du domaine ldap.local.

6. SECURITE

6.1 Durcissement de la VM Ubuntu

- Un utilisateur administrateur dédié (adminedj) est créé avec les droits sudo
- La connexion SSH root directe est désactivée
- UFW est activé avec des règles restrictives limitant les accès aux ports des services déployés

6.2 Gestion des secrets

- Les mots de passe et paramètres sensibles sont centralisés dans le fichier .env
- Le fichier .env est exclu du versionnement GitHub via .gitignore
- Un fichier .env.example (sans les valeurs sensibles) est versionné comme référence

6.3 VPN WireGuard

L'accès aux interfaces web internes (phpLDAPadmin, etc.) depuis l'extérieur du réseau local est sécurisé via le tunnel VPN WireGuard. Les interfaces web ne sont pas directement exposées sur Internet.

6.4 ClamAV

ClamAV assure la détection des malwares sur les fichiers présents sur le serveur. Le service tourne en conteneur isolé, accessible uniquement sur le réseau interne Docker.

7. DEPLOIEMENT ET CONFIGURATION

Pré-requis :

- Oracle VirtualBox 7.x installé
- Vagrant 2.4+ installé
- Box Vagrant ubuntu/jammy64 disponible

Procédure de déploiement :

1. Cloner le dépôt GitHub : git clone https://github.com/delcoco95/Securite_Serveur_Mediaschool.git

2. Copier le fichier .env.example vers .env et renseigner les valeurs (mots de passe, domaine LDAP)

3. Lancer le provisionnement : vagrant up

4. Vagrant installe automatiquement Docker, Docker Compose et lance les services

5. Vérifier l'état des conteneurs : vagrant ssh puis docker compose ps

Commandes de gestion de la VM :

- Démarrer : vagrant up
- Arrêter : vagrant halt
- Reconstruire depuis zéro : vagrant destroy -f && vagrant up
- Accéder en SSH : vagrant ssh

Commandes de gestion des conteneurs (depuis la VM) :

- Lister les conteneurs : docker compose ps
- Démarrer les services : docker compose up -d
- Arrêter les services : docker compose down

8. TESTS ET VALIDATION

Tests réalisés :

- Provisionnement complet de la VM via vagrant up sans erreur : succès
- Démarrage de l'ensemble des conteneurs Docker Compose : succès
- Accès à l'interface wg-easy depuis le navigateur hôte (http://192.168.56.10:51821) : succès
- Accès à phpLDAPadmin (http://192.168.56.10:8080) et connexion avec les identifiants LDAP : succès
- Création d'un utilisateur dans OpenLDAP via phpLDAPadmin : succès
- Service ClamAV accessible sur le port 3310 : succès
- Vérification des règles UFW actives : succès
- Test de connexion SSH depuis le poste hôte via l'IP 192.168.56.10 : succès