Dossier d'Exploitation pour le Veolia Cyber Hub (VCH)

Le Veolia Cyber Hub (VCH) est une solution de sécurisation positionnée en DMZ des sites industriels de Veolia Eau France. Ce dossier présente l'architecture technique et les solutions logicielles conteneurisées pour répondre aux exigences du projet.

Architecture système

Conformément aux exigences, nous utilisions Debian 12.9.0 avec une installation minimale pour réduire la surface d’attaque.

Installation de Docker

L'installation de Docker sera effectuée en suivant strictement le guide officiel :

```
# Mise à jour du système
apt-get update
apt-get install -y ca-certificates curl gnupg

# Ajout du dépôt Docker
install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg
chmod a+r /etc/apt/keyrings/docker.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null

# Installation de Docker

apt-get update
apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Services conteneurisés

Pour le service NTP, nous utiliserons l'image cturra/ntp

Configuration Docker Compose :

```
version: '3.9'

services:
  ntp:
    image: cturra/ntp:latest
    container_name: vch-ntp
```

```
    restart: always
    ports:
      - "123:123/udp"
    environment:
      - NTP_SERVERS=10.0.0.42,10.0.1.42
      - LOG_LEVEL=0
    cap_add:
      - SYS_TIME
      - SYS_NICE
    volumes:
      - ./ntp/config:/etc/chrony
      - ./ntp/logs:/var/log/chrony
```

Cette configuration permet de mettre en place :

- La synchronisation avec les serveurs NTP de Veolia
- L'exposition du port 123/UDP pour les équipements industriels
- La persistance des configurations et des journaux

Pour la journalisation, nous utiliserons une pile ELK (Elasticsearch, Logstash, Kibana) officielle, qui offre une solution complète pour la collecte, le stockage et la visualisation des journaux.

Configuration Docker Compose :

```
version: '3.9'

services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:8.12.0
    container_name: vch-elasticsearch
    environment:
      - node.name=vch-elasticsearch
      - discovery.type=single-node
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms2g -Xmx2g"
      - xpack.security.enabled=true
      - ELASTIC_PASSWORD=${ELASTIC_PASSWORD}
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - ./elasticsearch/data:/usr/share/elasticsearch/data
    restart: always

  logstash:
    image: docker.elastic.co/logstash/logstash:8.12.0
    container_name: vch-logstash
    ports:
      - "514:514/udp"
      - "514:514/tcp"
    volumes:
```

```
      - ./logstash/config/logstash.yml:/usr/share/logstash/config/logstash.yml:ro
      - ./logstash/pipeline:/usr/share/logstash/pipeline:ro
    environment:
      - "LS_JAVA_OPTS=-Xms1g -Xmx1g"
    restart: always
    depends_on:
      - elasticsearch

  kibana:
    image: docker.elastic.co/kibana/kibana:8.12.0
    container_name: vch-kibana
    ports:
      - "8514:5601"
    environment:
      - ELASTICSEARCH_URL=http://elasticsearch:9200
      - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
      - ELASTICSEARCH_USERNAME=kibana_system
      - ELASTICSEARCH_PASSWORD=${KIBANA_PASSWORD}
    restart: always
    depends_on:
      - elasticsearch

  forwarder:
    image: docker.elastic.co/logstash/logstash:8.12.0
    container_name: vch-forwarder
    volumes:
      - ./forwarder/config/logstash.yml:/usr/share/logstash/config/logstash.yml:ro
      - ./forwarder/pipeline:/usr/share/logstash/pipeline:ro
      - ./forwarder/certs:/usr/share/logstash/certs:ro
    environment:
      - "LS_JAVA_OPTS=-Xms512m -Xmx512m"
      - ESOC_IP_PRIMARY=10.0.0.80
      - ESOC_IP_SECONDARY=10.0.1.82
      - ESOC_PORT=6514
    restart: always
    depends_on:
      - elasticsearch
```

Configuration spécifique pour Logstash (pipeline) :

- Collecte des logs sur le port 514 (UDP/TCP)
- Stockage dans Elasticsearch avec une politique de rétention de 1 mois
- Transmission sécurisée vers l'E-SOC via TLS sur le port 6514

Pour le service de conformité, nous développerons une solution personnalisée basée sur des images officielles.

Configuration Docker Compose :

```
version: '3.9'

services:
  api:
```

```
    image: node:18-alpine
    container_name: vch-conformity-api
    volumes:
      - ./api:/app
    working_dir: /app
    command: npm start
    restart: always
    ports:
      - "3000:3000"

  frontend:
    image: nginx:alpine
    container_name: vch-conformity-frontend
    volumes:
      - ./frontend:/usr/share/nginx/html
      - ./frontend/nginx.conf:/etc/nginx/conf.d/default.conf
    ports:
      - "8443:443"
    restart: always
    depends_on:
      - api

  mongodb:
    image: mongo:6
    container_name: vch-conformity-db
    environment:
      - MONGO_INITDB_ROOT_USERNAME=${MONGO_ROOT_USER}
      - MONGO_INITDB_ROOT_PASSWORD=${MONGO_ROOT_PASSWORD}
    volumes:
      - ./mongodb/data:/data/db
    restart: always

  forwarder:
    image: node:18-alpine
    container_name: vch-conformity-forwarder
    volumes:
      - ./forwarder:/app
    working_dir: /app
    command: npm start
    environment:
      - OCI_PRIMARY_IP=10.0.5.42
      - OCI_SECONDARY_IP=10.0.6.42
      - OCI_PORT=443
    restart: always
    depends_on:
      - api
```

Pour l'agent de collecte Linux, nous avons développé un script Bash conforme aux recommandations de l'ANSSI qui vérifiera :

- La configuration des comptes et des droits
- Les services actifs et leurs configurations
- Les paramètres de durcissement du système

- Les mises à jour de sécurité

Pour le service proxy, nous utiliserons HAProxy, une solution robuste et légère pour le relais de flux.

Configuration Docker Compose :

```
version: '3.9'

services:
  haproxy:
    image: haproxy:2.8-alpine
    container_name: vch-proxy
    ports:
      - "8530:8530"
      - "8080:8080"
      - "53:53/udp"
      - "53:53/tcp"
    volumes:
      - ./haproxy/haproxy.cfg:/usr/local/etc/haproxy/haproxy.cfg:ro
      - ./haproxy/logs:/var/log/haproxy
    restart: always
```

Configuration HAProxy pour les différents relais :
- Relais WSUS (port 8530) vers 10.0.11.42:8530
- Relais Web (port 8080) vers 10.0.22.42:8080
- Relais DNS (port 53) vers 10.0.15.42:53

Sécurisation de la solution

- Mise à jour régulière du système par des passage d'agents, planifiées sur les différentes sites
- Configuration d'un pare-feu avec iptables
- Désactivation des services inutiles
- Configuration de fail2ban pour la protection SSH
- Audit régulier (en même temps que les mises à jour)

Sécurisation de Docker

- Utilisation exclusive d'images officielles ou vérifiées comme précisé dans le brief
- Limitation des capacités des conteneurs
- Utilisation de réseaux Docker isolés
- Montage des volumes en lecture seule quand possible

Administration à distance sécurisée

- Accès SSH avec authentification par clé uniquement
- Utilisation de la DMZ pour un accès sécurisé

- Journalisation de toutes les actions d’administration

Surveillance et maintenance

Mise en place d'une solution de supervision basée sur un script python pour surveiller :

- La liste des postes, qui respectent la politique de mots de passe Veolia Eau France
- La liste des groupes, des comptes et leur typologie (administrateur/utilisateur) ainsi que leur présence sur chaque machine
- Les équipements qui n’ont pas envoyé de données sur les dernières 24h doit être disponible

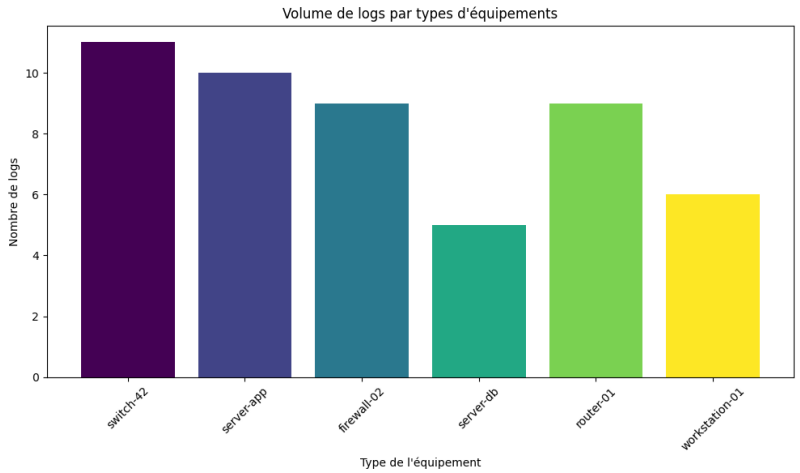
Procédure de mise à jour sécurisée :

1. Scan de sécurité des nouvelles images
2. Test en environnement de pré-production
3. Déploiement avec possibilité de rollback

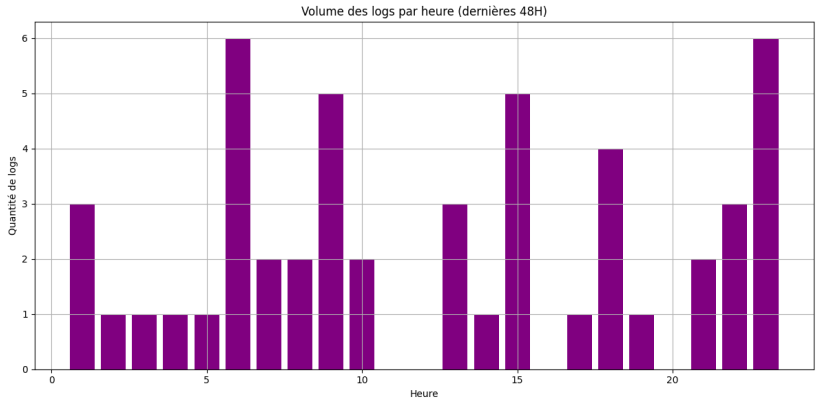
Conclusion

Cette solution répond aux exigences du projet en proposant une architecture conteneurisée, sécurisée et minimale pour le Veolia Cyber Hub. L'utilisation exclusive d'images Docker officielles ou vérifiées, combinée à un système hôte durci, permet d'assurer un niveau de sécurité optimal pour les services critiques de Veolia Eau France.

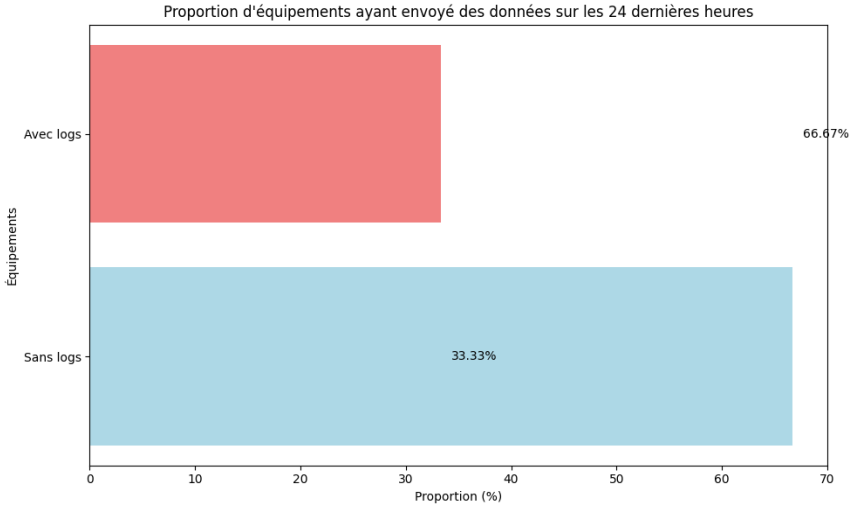
Annexes :



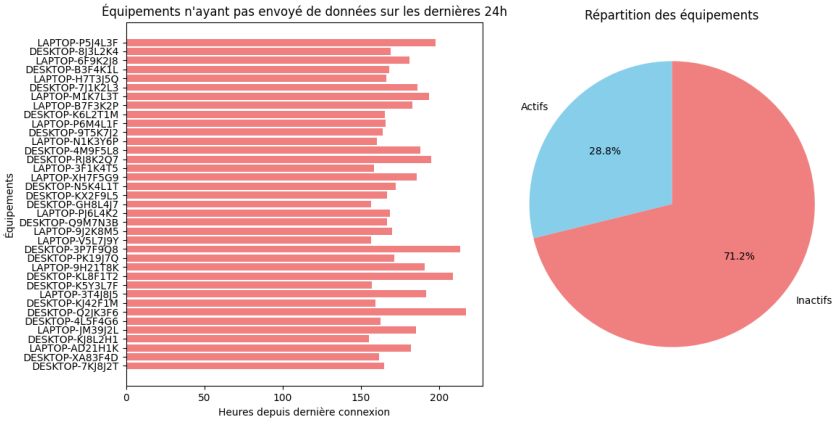
Chaque barre correspond à un équipement et sa hauteur indique la quantité de logs générés.



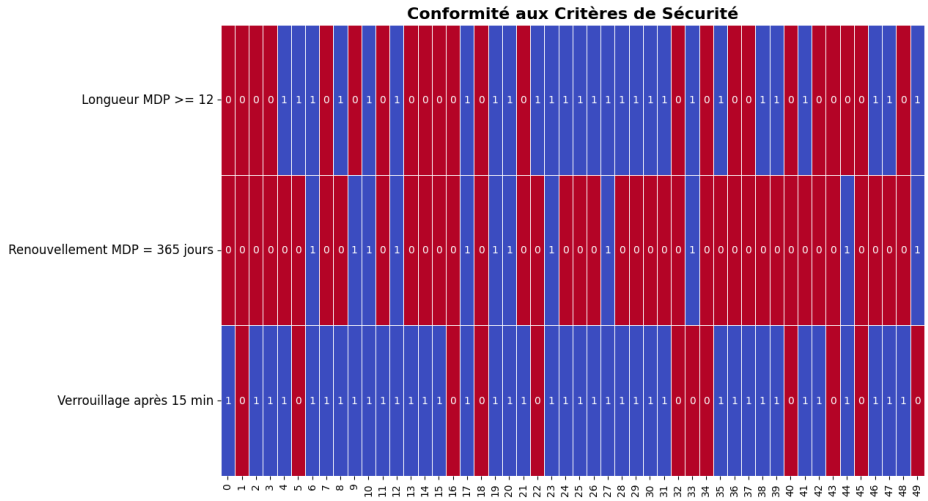
Graphe représentant le volume des logs générés par heure sur les dernières 48 heures

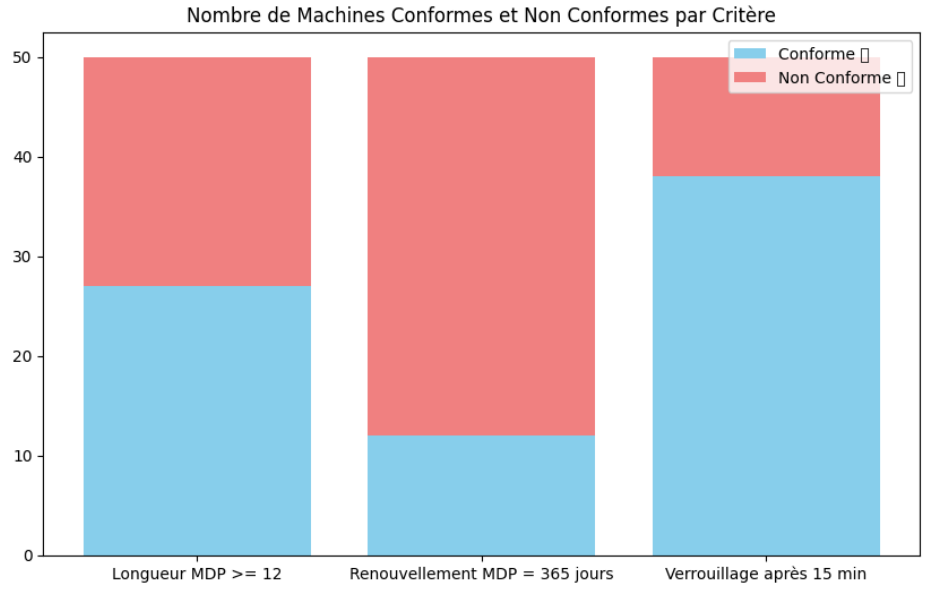


Graphe représentant la proportion d’équipements ayant envoyé des logs au cours des dernières 24 heures

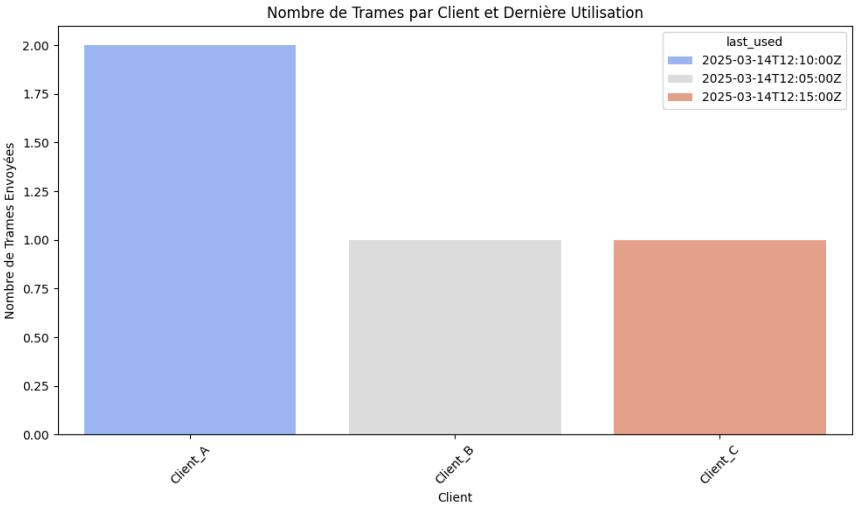


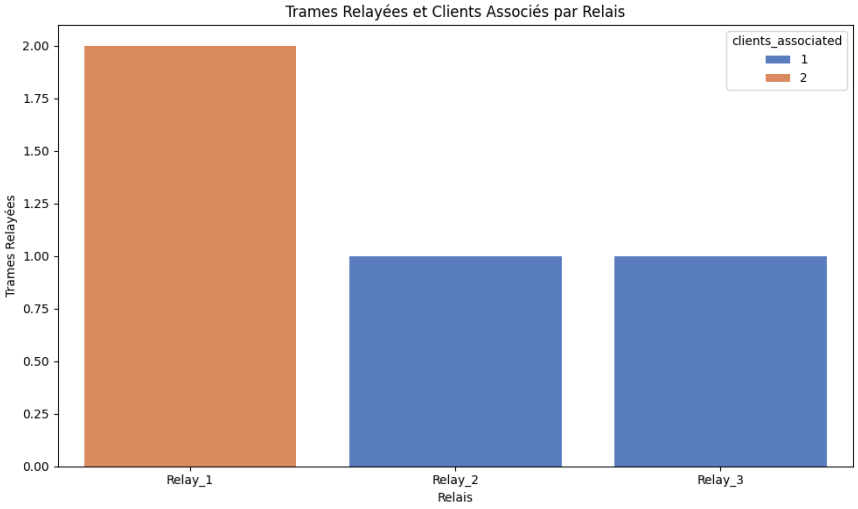
Le premier graphe représente le temps écoulé depuis la dernière connexion des équipements inactifs, tandis que le second illustre la répartition des équipements actifs et inactifs au cours des dernières 24 heures.





Graphe représentant la conformité des machines aux critères de sécurité





Graphique des trames clients et relayées.