Lycée Saint Adjutor — BTS SIO 2e année

Alert'Intrusion

Documentation technique — Déploiement Wazuh

Projet	Alert'Intrusion
Version Wazuh	4.14.4
OS serveur	Ubuntu 22.04 LTS
IP Manager	172.20.100.6
Date de présentation	7 avril 2026

1. Contexte et objectifs

Le projet Alert'Intrusion répond à deux besoins identifiés lors d'un audit de sécurité sur l'infrastructure de l'entreprise X :

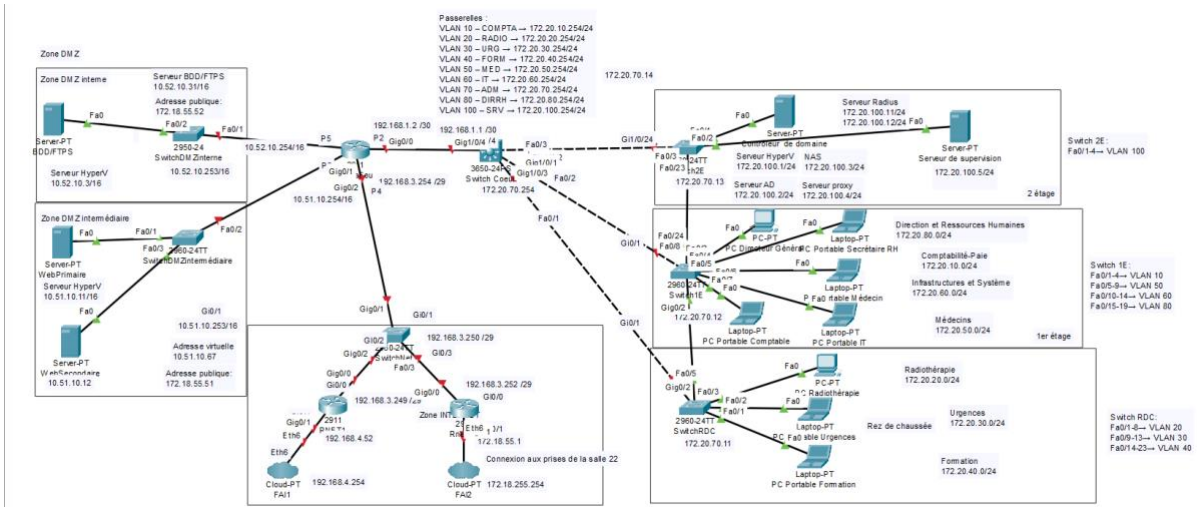
- Détection d'intrusions sur la zone DMZ (serveur web hébergeant l'application de commandes/réservations clients)
- Centralisation des journaux des équipements LAN (switchs, serveurs Linux et Windows)
- Mise en place d'un proxy SSL (Stormshield SN160) pour filtrage HTTPS et protection antivirale

Wazuh couvre les points 1 et 2 : c'est un SIEM/HIPS open-source qui combine la collecte de journaux, la détection d'anomalies et la corrélation d'alertes.

2. Architecture de la solution

Composant	IP / Port	Rôle
Wazuh Manager	172.20.100.6	Réception et analyse des logs
Wazuh Indexer (OpenSearch)	:9200	Stockage et indexation des alertes
Wazuh Dashboard	:443 (HTTPS)	Interface de visualisation
Agents Linux	TCP 1514/1515	Collecte logs serveurs Linux
Agent Windows	TCP 1514/1515	Collecte logs Windows EventLog
Équipements Cisco	UDP 514 (syslog)	Pas d'agent natif — syslog uniquement

> *Les équipements Cisco IOS ne supportent pas l'agent Wazuh. La seule méthode d'intégration est l'envoi de syslog UDP vers le Manager sur le port 514.*



3. Installation du Manager Wazuh (Ubuntu 22.04)

3.1 Pré-requis

Ressource	Minimum recommandé
RAM	4 Go (réduire le heap JVM à 1 Go si contrainte)
CPU	2 vCPU
Disque	50 Go minimum
OS	Ubuntu 22.04 LTS (testé)
Accès réseau	DNS fonctionnel vers packages.wazuh.com

3.2 Ajout du dépôt Wazuh

Si le DNS est absent, ajouter temporairement : echo "nameserver 8.8.8.8" >> /etc/resolv.conf

```
apt-get install gnupg apt-transport-https

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH \
  | gpg --no-default-keyring \
  --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg \
  --import

chmod 644 /usr/share/keyrings/wazuh.gpg

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] \
```

```
  https://packages.wazuh.com/4.x/apt/ stable main" \
  | tee /etc/apt/sources.list.d/wazuh.list

apt-get update
```

3.3 Installation via le script officiel

Téléchargement du script (version 4.14) :

```
curl -sO https://packages.wazuh.com/4.14/wazuh-install.sh
curl -sO https://packages.wazuh.com/4.14/config.yml
```

Éditer config.yml — remplacer toutes les occurrences d'IP par 172.20.100.6 :

```
nodes:
  indexer:
    - name: node-1
      ip: "172.20.100.6"
  server:
    - name: wazuh-1
      ip: "172.20.100.6"
  dashboard:
    - name: dashboard
      ip: "172.20.100.6"
```

> *Ne pas utiliser 127.0.0.1 même en déploiement mono-noeud. Les certificats TLS sont générés avec l'IP indiquée. Un mismatch cassera la communication avec les agents.*

Lancer l'installation (une commande à la fois, dans l'ordre) :

```
bash wazuh-install.sh --generate-config-files
bash wazuh-install.sh --wazuh-indexer node-1
bash wazuh-install.sh --start-cluster
bash wazuh-install.sh --wazuh-server wazuh-1
bash wazuh-install.sh --wazuh-dashboard dashboard
```

Récupérer les credentials générés :

```
tar -O -xvf wazuh-install-files.tar \
  wazuh-install-files/wazuh-passwords.txt
```

3.4 Vérification post-installation

```
systemctl status wazuh-manager wazuh-indexer wazuh-dashboard

# Vérifier que les ports écoutent
ss -tlnp | grep -E '443|9200|55000|1514|1515'
```

```
# Tester l'API
curl -k -u wazuh-wui:<PASSWORD> https://localhost:55000
```

Le dashboard est accessible sur : https://172.20.100.6 (login : admin)

4. Configuration du Manager

Fichier principal : /var/ossec/etc/ossec.conf

4.1 Configuration de base

```
<ossec_config>
  <global>
    <logall>yes</logall>
    <logall_json>yes</logall_json>
  </global>

  <alerts>
    <log_alert_level>3</log_alert_level>
  </alerts>

  <!-- Réception agents (Linux/Windows) -->
  <remote>
    <connection>secure</connection>
    <port>1514</port>
    <protocol>tcp</protocol>
  </remote>

  <!-- Réception syslog Cisco -->
  <remote>
    <connection>syslog</connection>
    <port>514</port>
    <protocol>udp</protocol>
    <allowed-ips>172.20.0.0/16</allowed-ips>
  </remote>
</ossec_config>
```

Après toute modification :

```
systemctl restart wazuh-manager
```

4.2 Ouverture des ports firewall

```
ufw allow 1514/tcp   # Agents
ufw allow 1515/tcp   # Enrollment
ufw allow 514/udp    # Syslog Cisco
ufw allow 443/tcp    # Dashboard
```

5. Déploiement de l'agent Linux

5.1 Installation (Debian/Ubuntu)

Si la machine a accès internet :

```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH \
  | gpg --dearmor -o /usr/share/keyrings/wazuh.gpg

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] \
  https://packages.wazuh.com/4.x/apt/ stable main" \
  > /etc/apt/sources.list.d/wazuh.list

apt update && WAZUH_MANAGER="172.20.100.6" apt install wazuh-agent
```

Si la machine n'a pas d'accès DNS (maquette isolée) — méthode offline :

```
# Sur le manager : récupérer le .deb en cache
find /var/cache/apt/archives/ -name 'wazuh-agent*'
scp /var/cache/apt/archives/wazuh-agent_*.deb admin@<IP_CIBLE>:/tmp/

# Sur la cible :
WAZUH_MANAGER="172.20.100.6" dpkg -i /tmp/wazuh-agent_*.deb
```

5.2 Enregistrement et démarrage

```
/var/ossec/bin/agent-auth -m 172.20.100.6

systemctl daemon-reload
systemctl enable --now wazuh-agent
systemctl status wazuh-agent
```

5.3 Vérification

```
# Côté agent
tail -20 /var/ossec/logs/ossec.log

# Côté manager
/var/ossec/bin/manage_agents -l
```

6. Déploiement de l'agent Windows

6.1 Installation via PowerShell (en admin)

```
# Télécharger le MSI
Invoke-WebRequest -Uri "https://packages.wazuh.com/4.x/windows/wazuh-agent-
4.14.4-1.msi" \
  -OutFile "$env:TEMP\wazuh-agent.msi"

# Vérifier que le fichier est complet (~25 Mo)
(Get-Item "$env:TEMP\wazuh-agent.msi").Length

# Installation silencieuse avec logs
msiexec /i "$env:TEMP\wazuh-agent.msi" /l*v "C:\wazuh-install.log" `
  WAZUH_MANAGER="172.20.100.6" `
  WAZUH_REGISTRATION_SERVER="172.20.100.6" `
  WAZUH_AGENT_NAME="WIN-LEIDOSCOPE-01"
```

6.2 Enregistrement et démarrage

```
# Enregistrement manuel si nécessaire
& "C:\Program Files (x86)\ossec-agent\agent-auth.exe" -m 172.20.100.6

# Démarrer le service
NET START WazuhSvc

# Vérifier
Get-Service WazuhSvc
```

6.3 Diagnostic si 'Never Connected'

```
# Tester la connectivité vers le manager
Test-NetConnection -ComputerName 172.20.100.6 -Port 1515
# TcpTestSucceeded doit être True

# Logs de l'agent
Get-Content "C:\Program Files (x86)\ossec-agent\ossec.log" -Tail 30

# Si DNS absent sur Windows
Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses
"8.8.8.8"
```

7. Intégration des équipements Cisco (Syslog)

> *Cisco IOS n'a pas d'agent Wazuh natif. L'intégration passe par l'envoi de syslog UDP vers le port 514 du Manager. Les logs apparaissent dans le dashboard sous l'agent 'wazuh' (le Manager lui-même).*

7.1 Configuration sur le switch/routeur IOS

Important : toutes les commandes doivent être saisies en mode configure terminal (#conf t), pas en mode exec (#).

```
configure terminal
service timestamps log datetime msec localtime
logging host 172.20.100.6
logging trap informational
logging facility local7
logging source-interface Vlan100
end
write memory
show logging
```

> *La commande 'logging host <IP> transport udp port 514' n'existe pas sur IOS 15.0 (C2960). La syntaxe simplifiée 'logging host <IP>' envoie en UDP 514 par défaut.*

7.2 Vérification côté switch

La commande show logging doit afficher :

```
Trap logging: level informational
    Logging to 172.20.100.6 (udp port 514, ...)
```

7.3 Vérification côté Manager

```
# Vérifier que le port 514 écoute
ss -ulnp | grep 514
# Doit afficher : UNCONN ... 0.0.0.0:514 ... wazuh-remoted

# Capturer les paquets en temps réel
tcpdump -i any udp port 514 -nn -c 20
```

7.4 Diagnostic si les logs n'arrivent pas

Causes classiques et corrections :

Symptôme	Action
tcpdump ne voit rien	Vérifier le routage entre le switch et 172.20.100.6 (ping depuis le switch)
Port 514 absent de ss -ulnp	Ajouter le bloc <remote> syslog dans ossec.conf + restart manager
Logs absents du dashboard	Activer logall dans ossec.conf, créer l'index wazuh-archives-* dans le dashboard
ACL bloque UDP 514	Vérifier l'ACL sur l'interface source du switch vers 172.20.100.6

8. Règles de détection

8.1 Règles intégrées (actives par défaut)

Wazuh charge automatiquement toutes les règles depuis /var/ossec/ruleset/rules/. Les principales :

- 0040-sshd_rules.xml — Brute force SSH, connexions root
- 0580-win-security_rules.xml — Windows EventLog, RDP, privilege escalation
- 0095-syslog_rules.xml — Syslog générique (Cisco inclus)

Le niveau minimum d'alerte est contrôlé par :

```
# Dans /var/ossec/etc/ossec.conf
<alerts>
  <log_alert_level>3</log_alert_level>
</alerts>
```

8.2 Règles custom (local_rules.xml)

Fichier : /var/ossec/etc/rules/local_rules.xml

```
<group name="local,leidoscope">

  <!-- SSH : brute force (5 échecs en 60s) -->
```

```
  <rule id="100001" level="10" frequency="5" timeframe="60">
    <if_matched_sid>5716</if_matched_sid>
    <description>Brute force SSH - 5 echecs en 60s</description>
    <group>authentication_failures,</group>
  </rule>

  <!-- SSH : connexion root -->
  <rule id="100002" level="12">
    <if_sid>5715</if_sid>
    <match>root</match>
    <description>Connexion SSH en tant que root</description>
  </rule>

  <!-- Linux : sudo utilise -->
  <rule id="100003" level="6">
    <if_sid>5402</if_sid>
    <description>Utilisation de sudo detectee</description>
  </rule>

  <!-- Windows : brute force (5 echecs en 60s) -->
  <rule id="100004" level="10" frequency="5" timeframe="60">
    <if_matched_sid>60122</if_matched_sid>
    <description>Brute force Windows detecte</description>
    <group>authentication_failures,</group>
  </rule>

  <!-- Cisco : changement de configuration -->
  <rule id="100006" level="8">
    <if_sid>4700</if_sid>
    <match>CONFIG_I</match>
    <description>Configuration Cisco modifiee</description>
  </rule>

  <!-- Cisco : interface down -->
  <rule id="100007" level="6">
    <if_sid>4700</if_sid>
    <match>UPDOWN</match>
    <match>changed state to down</match>
    <description>Interface Cisco passee down</description>
  </rule>

  <!-- Cisco : echec de connexion -->
  <rule id="100008" level="10">
    <if_sid>4700</if_sid>
    <match>LOGIN_FAILED</match>
    <description>Echec connexion equipement Cisco</description>
    <group>authentication_failures,</group>
  </rule>

</group>
```

Validation de la syntaxe et redémarrage :

```
/var/ossec/bin/wazuh-logtest
systemctl restart wazuh-manager
```

9. Jeu de tests

9.1 Test SSH brute force (Linux)

Depuis le Manager ou une autre machine Linux :

```
apt install sshpass -y

for i in $(seq 1 20); do
  sshpass -p 'wrongpassword' ssh -o StrictHostKeyChecking=no \
    root@172.20.100.7 2>/dev/null
done
```

Résultat attendu dans le dashboard (Security operations > Events) : alertes de niveau 10+ avec tactique MITRE ATT&CK T1110 (Brute Force) dans Credential Access.

9.2 Tests Nmap (requis par le cahier des charges)

```
# Depuis une machine sur le LAN
nmap -sS 172.20.100.7    # SYN scan — détecte les ports TCP ouverts
nmap -sU 172.20.100.7    # UDP scan
nmap -sV 172.20.100.7    # Version des services
```

Wazuh détecte les scans via les règles intégrées et les mappe sous MITRE T1046 (Network Service Discovery) dans la catégorie Discovery.

9.3 Test déni de service avec hping3

```
# Simulation SYN flood vers le serveur web DMZ
hping3 -S --flood -V -p 80 <IP_SERVEUR_DMZ>

# Stopper avec Ctrl+C
```

9.4 Tests Linux divers

```
# Tentative d'escalade de privilèges
su - wronguser

# Modification d'un fichier système critique
echo 'test' >> /etc/passwd
```

```
# Création d'un utilisateur non autorisé
useradd hackerman
```

9.5 Test Cisco (génération de logs)

```
configure terminal
no logging host 172.20.100.6
logging host 172.20.100.6
end
# Déclenche CONFIG_I dans Wazuh
```

9.6 Vérification dans le dashboard

- Menu : Security operations > Events
- Filtrer par niveau : rule.level >= 5
- Filtrer par agent : agent.name: <nom_agent>
- Vérifier la plage de temps (haut à droite) : mettre Last 24 hours si rien n'apparaît

Tester manuellement un log SSH sans agent via wazuh-logtest :

```
/var/ossec/bin/wazuh-logtest
# Coller :
Mar 10 01:02:02 linux sshd[1234]: Failed password for root from 192.168.1.1
port 22 ssh2
```

10. Diagnostic et dépannage

10.1 Commandes de base

```
# Statut des services
systemctl status wazuh-manager wazuh-indexer wazuh-dashboard

# Lister les agents connectés
/var/ossec/bin/manage_agents -l

# Logs principaux
tail -50 /var/ossec/logs/ossec.log
journalctl -u wazuh-indexer -f
journalctl -u wazuh-dashboard -f
```

10.2 Problèmes connus et solutions

Erreur	Solution
Dashboard : 'server not ready yet'	Redémarrer l'indexer d'abord, attendre 30s, puis restart dashboard
Indexer crash : log dir missing	mkdir -p /var/log/wazuh-indexer && chown -R wazuh-indexer:wazuh-indexer /var/log/wazuh-indexer
Agent 'Never Connected'	Relancer agent-auth -m 172.20.100.6, vérifier TCP 1515 avec Test-NetConnection
JVM crash (heap trop grand)	Réduire dans /etc/wazuh-indexer/jvm.options : -Xms1g -Xmx1g
API 401 Unauthorized	Vérifier le mot de passe wazuh-wui dans wazuh-install-files.tar
Syslog Cisco absent du dashboard	Activer logall_json dans ossec.conf, créer l'index wazuh-archives-* dans Dashboard Management

10.3 Ordre de démarrage correct

```
systemctl restart wazuh-indexer
# Attendre ~30s
curl -k -u admin:admin https://localhost:9200/_cluster/health?pretty
# Vérifier status 'green' ou 'yellow'
systemctl restart wazuh-manager
systemctl restart wazuh-dashboard
```

11. Correspondances MITRE ATT&CK

ID	Tactique	Scénario de test associé
T1110	Credential Access	Script SSH brute force / erreurs Windows
T1046	Discovery	Scan nmap -sS / -sU / -sV
T1498	Impact	SYN flood hping3 vers le serveur web DMZ
T1078	Lateral Movement	Tentative de connexion root SSH

Le mapping MITRE est automatique — Wazuh fait correspondre ses règles aux tactiques sans configuration manuelle.

12. Références

- Documentation officielle Wazuh 4.14 : https://documentation.wazuh.com/current/
- Règles intégrées : /var/ossec/ruleset/rules/
- Règles custom : /var/ossec/etc/rules/local_rules.xml
- Configuration principale : /var/ossec/etc/ossec.conf
- Logs : /var/ossec/logs/ossec.log et /var/ossec/logs/alerts/

> *Ce document couvre les activités A6, A7, A10, A11, A13 et A14 du cahier des charges Alert'Intrusion.*