# Mise en place d'un système de sécurité réseau

Déploiement d'un conteneur LXC avec Fail2ban et Portsentry sur Proxmox

**Auteur :** Ahmed Koucha

**Organisation :** GSB

**Date :** Mai 2025

## Table des matières



# 1. Introduction

## 1.1 Contexte du projet

Dans le cadre de la sécurisation de l'infrastructure réseau de GSB, ce projet vise à mettre en place un système de détection et prévention d'intrusions basé sur un conteneur LXC hébergé sur Proxmox. L'objectif est de détecter et bloquer automatiquement les tentatives d'attaques par force brute et les scans de ports malveillants.

### 1.2 Objectifs

- Déployer un conteneur LXC dédié à la sécurité réseau
- Installer et configurer Fail2ban pour la détection et le blocage des attaques par force brute
- Mettre en place Portsentry pour la détection et le blocage des scans de ports
- Intégrer ces outils avec le pare-feu système
- Mettre en place un système de journalisation et d'alerte

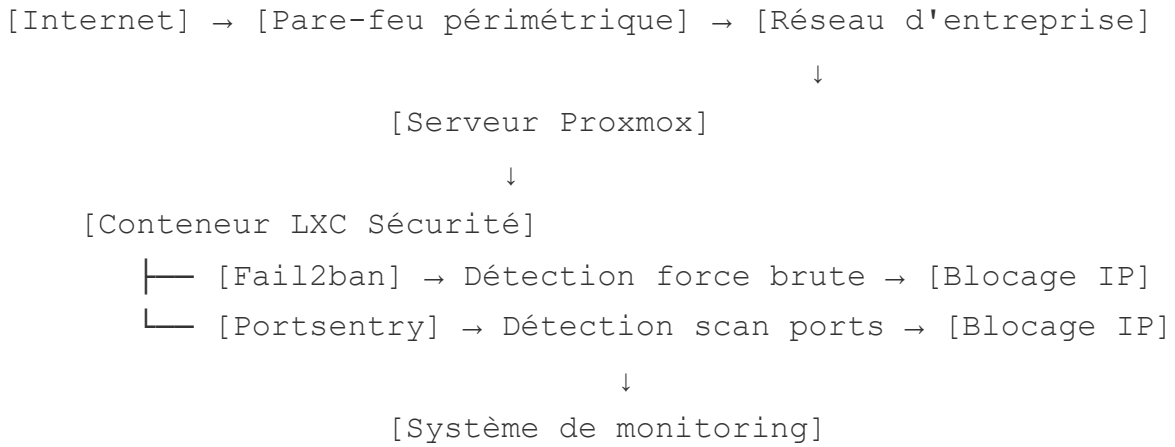
## 2. Architecture de la solution

### 2.1 Vue d'ensemble

La solution repose sur un conteneur LXC dédié à la sécurité, hébergé sur l'infrastructure Proxmox existante. Ce conteneur exécute deux services principaux :

- **Fail2ban** : Analyse les journaux système pour détecter les tentatives d'authentification échouées et bloque les adresses IP sources
- **Portsentry** : Surveille les ports réseau pour détecter les activités de scan et bloque les adresses IP suspectes

### 2.2 Schéma d'architecture



[Alertes administrateurs]

# 3. Préparation de l'environnement

## 3.1 Création du conteneur LXC

```
# Création d'un conteneur basé sur Debian 11
pct create 100 local:vztmpl/debian-11-standard_11.0-1_amd64.tar.gz \
  --hostname security-container \
  --cores 2 \
  --memory 2048 \
  --swap 1024 \
  --rootfs local-lvm:20 \
  --net0 name=eth0,bridge=vmbr0,ip=10.31.224.50/24,gw=10.31.227.254 \
  --onboot 1 \
  --unprivileged 0
```

## 3.2 Configuration réseau du conteneur

```
# Démarrage du conteneur
pct start 100

# Accès au conteneur
pct enter 100

# Vérification de la configuration réseau
ip addr show
ping -c 4 google.com
```

## 3.3 Mise à jour du système

```
apt update
apt upgrade -y
apt install -y vim wget curl net-tools iptables
```

# 4. Installation et configuration de Fail2ban

## 4.1 Installation

```
apt install -y fail2ban
```

## 4.2 Configuration de base

```
# Création du fichier de configuration local
cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local

# Édition du fichier de configuration
vim /etc/fail2ban/jail.local
```

Contenu de base à modifier dans jail.local :

```
[DEFAULT]
# Temps de bannissement en secondes (12 heures)
bantime = 43200
# Nombre d'échecs avant bannissement
maxretry = 3
# Période d'observation en secondes
findtime = 600
# Ignorer les adresses IP locales
ignoreip = 127.0.0.1/8 10.31.224.0/24 10.31.232.0/24

# Activation de la détection SSH
[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
```

## 4.3 Configuration des filtres personnalisés

```
# Création d'un filtre personnalisé pour les tentatives d'authentification we
vim /etc/fail2ban/filter.d/web-auth.conf
```

Contenu du filtre web-auth.conf :

```
[Definition]
failregex = ^ -.*"(GET|POST).*(login|admin|wp-login).* 401
            ^ -.*"(GET|POST).*(login|admin|wp-login).* 403
ignoreregex =
```

## 4.4 Configuration des actions

```
# Création d'une action personnalisée pour la notification
vim /etc/fail2ban/action.d/custom-notify.conf
```

Contenu de custom-notify.conf :

```
[Definition]
actionstart =
actionstop =
actioncheck =
actionban = echo "IP:  a été bannie pour  à $(date)" | mail -s "Fail2ban: IP
actionunban =

[Init]
```

## 4.5 Activation des jails

```
vim /etc/fail2ban/jail.d/custom.conf
```

Contenu de custom.conf :

```
[web-auth]
enabled = true
filter = web-auth
logpath = /var/log/nginx/access.log
maxretry = 5
bantime = 86400
action = iptables-multiport[name=web-auth, port="http,https"]
         custom-notify
```

## 4.6 Démarrage et activation du service

```
systemctl enable fail2ban
systemctl start fail2ban
systemctl status fail2ban
```

# 5. Installation et configuration de Portsentry

## 5.1 Installation

```
apt install -y portsentry
```

## 5.2 Configuration de base

```
# Sauvegarde de la configuration d'origine
cp /etc/portsentry/portsentry.conf /etc/portsentry/portsentry.conf.bak

# Édition du fichier de configuration
vim /etc/portsentry/portsentry.conf
```

Modifications importantes dans portsentry.conf :

```
# Passer du mode basic au mode avancé
ADVANCED_MODE_TCP="1"
ADVANCED_MODE_UDP="1"

# Activer la réponse automatique
BLOCK_UDP="1"
BLOCK_TCP="1"

# Configuration du mode de blocage
KILL_ROUTE="/sbin/iptables -I INPUT -s $TARGET$ -j DROP"
```

## 5.3 Configuration des ports à surveiller

```
vim /etc/portsentry/portsentry.conf
```

Modification de la section des ports :

```
# Ports TCP à surveiller en mode avancé
TCP_PORTS="1,11,15,79,111,119,143,540,635,1080,1524,2000,5742,6667,12345,123

# Ports UDP à surveiller en mode avancé
UDP_PORTS="1,7,9,69,161,162,513,635,640,641,700,37444,34555,31335,32770,3277
```

### 5.4 Configuration du mode de démarrage

```
vim /etc/default/portsentry
```

Modification du mode :

```
TCP_MODE="atcp"
UDP_MODE="audp"
```

### 5.5 Démarrage et activation du service

```
systemctl enable portsentry
systemctl restart portsentry
systemctl status portsentry
```

## 6. Intégration avec le pare-feu système

### 6.1 Configuration d'iptables

```
# Installation des outils de persistance iptables
apt install -y iptables-persistent

# Configuration des règles de base
iptables -F
iptables -X
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# Autoriser les connexions établies
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# Autoriser le trafic local
iptables -A INPUT -i lo -j ACCEPT

# Autoriser SSH (à adapter selon vos besoins)
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# Sauvegarder les règles
netfilter-persistent save
```

## 6.2 Synchronisation des listes de blocage

Création d'un script pour synchroniser les listes de blocage :

```
vim /usr/local/bin/sync-blocklists.sh
```

Contenu du script :

```
#!/bin/bash

# Récupération des adresses IP bloquées par Fail2ban
FAIL2BAN_IPS=$(fail2ban-client status | grep "Jail list" | sed "s/^.*Jail lis

# Récupération des adresses IP bloquées par Portsentry
PORTSENTRY_IPS=$(grep -E "^Blocked|^Detected" /var/lib/portsentry/portsentry.

# Vérification et ajout des règles iptables si nécessaire
for IP in $FAIL2BAN_IPS $PORTSENTRY_IPS; do
    if ! iptables -L INPUT -v -n | grep -q "$IP"; then
        echo "Ajout de l'IP $IP à la liste de blocage iptables"
        iptables -I INPUT -s "$IP" -j DROP
    fi
done

# Sauvegarde des règles
netfilter-persistent save
```

Rendre le script exécutable et planifier son exécution :

```
chmod +x /usr/local/bin/sync-blocklists.sh
```

```
# Ajout d'une tâche cron pour exécuter le script toutes les heures
echo "0 * * * * root /usr/local/bin/sync-blocklists.sh" > /etc/cron.d/sync-bl
```

# 7. Mise en place de la journalisation et des alertes

## 7.1 Configuration de la rotation des logs

```
vim /etc/logrotate.d/security-tools
```

Contenu du fichier :

```
/var/log/fail2ban.log {
    weekly
    rotate 4
    compress
    delaycompress
    missingok
    postrotate
        systemctl reload fail2ban
    endscript
}

/var/log/portsentry.log {
    weekly
    rotate 4
    compress
    delaycompress
    missingok
    create 0640 root adm
}
```

## 7.2 Configuration des alertes par email

```
# Installation de l'agent de transfert de mail
apt install -y postfix mailutils

# Configuration de base de Postfix (choisir "Internet Site" lors de l'instal
```

## 7.3 Script de rapport quotidien

```
vim /usr/local/bin/security-report.sh
```

Contenu du script :

```
#!/bin/bash

DATE=$(date +"%Y-%m-%d")
REPORT_FILE="/tmp/security-report-$DATE.txt"

echo "Rapport de sécurité du $DATE" > $REPORT_FILE
echo "================================" >> $REPORT_FILE
echo "" >> $REPORT_FILE

echo "1. Statistiques Fail2ban" >> $REPORT_FILE
echo "------------------------" >> $REPORT_FILE
fail2ban-client status | grep "Jail list" | sed "s/^.*Jail list://g" | sed "s
    echo "Jail: $jail" >> $REPORT_FILE
    fail2ban-client status "$jail" >> $REPORT_FILE
    echo "" >> $REPORT_FILE
done

echo "2. Statistiques Portsentry" >> $REPORT_FILE
echo "--------------------------" >> $REPORT_FILE
echo "Tentatives de scan bloquées aujourd'hui:" >> $REPORT_FILE
grep -E "^$(date +"%b %d")" /var/log/syslog | grep portsentry | grep "blocke
echo "" >> $REPORT_FILE

echo "3. Top 10 des adresses IP bloquées" >> $REPORT_FILE
echo "----------------------------------" >> $REPORT_FILE
iptables -L INPUT -n | grep DROP | awk '{print $4}' | sort | uniq -c | sort -

# Envoi du rapport par email
mail -s "Rapport de sécurité quotidien - $DATE" admin@gsb.org < $REPORT_FILE

# Nettoyage
rm $REPORT_FILE
```

Rendre le script exécutable et planifier son exécution :

```
chmod +x /usr/local/bin/security-report.sh

# Ajout d'une tâche cron pour exécuter le script tous les jours à 7h
echo "0 7 * * * root /usr/local/bin/security-report.sh" > /etc/cron.d/securit
```

# 8. Tests et validation

## 8.1 Test de Fail2ban

```
# Simulation de tentatives d'authentification échouées SSH
# Depuis une machine externe, essayer de se connecter plusieurs fois avec un

# Vérification du statut de Fail2ban
fail2ban-client status sshd
```

## 8.2 Test de Portsentry

```
# Depuis une machine externe, lancer un scan de ports
# Par exemple avec nmap : nmap -sS -p 1-1000

# Vérification des logs Portsentry
tail -f /var/log/syslog | grep portsentry
```

## 8.3 Vérification des règles iptables

```
# Affichage des règles de blocage
iptables -L INPUT -n | grep DROP
```

# 9. Maintenance et surveillance

## 9.1 Commandes utiles pour la surveillance

```
# Statut des services
systemctl status fail2ban
```

```
systemctl status portsentry

# Consultation des logs
tail -f /var/log/fail2ban.log
tail -f /var/log/syslog | grep portsentry

# Liste des adresses IP bannies par Fail2ban
fail2ban-client status | grep "Jail list" | sed "s/^.*Jail list://g" | sed "s
```

### 9.2 Procédure de déblocage d'une adresse IP

```
# Déblocage d'une adresse IP dans Fail2ban
fail2ban-client set  unbanip

# Déblocage d'une adresse IP dans iptables
iptables -D INPUT -s  -j DROP

# Suppression d'une adresse IP de la liste de blocage de Portsentry
vim /etc/portsentry/portsentry.blocked.tcp
# ou
vim /etc/portsentry/portsentry.blocked.udp
# Puis redémarrer Portsentry
systemctl restart portsentry
```

## 10. Conclusion

Cette solution de sécurité réseau basée sur Fail2ban et Portsentry offre une protection efficace contre les tentatives d'intrusion courantes. La mise en place dans un conteneur LXC dédié permet une isolation des services de sécurité et facilite la maintenance. L'automatisation des rapports et des alertes permet une surveillance proactive de la sécurité du réseau.

La solution s'intègre parfaitement dans la stratégie de défense en profondeur de l'entreprise et constitue une première ligne de défense efficace contre les tentatives d'intrusion courantes.

## 11. Annexes

### 11.1 Références

- Documentation officielle de Fail2ban : https://www.fail2ban.org/wiki/index.php/Main_Page

- Documentation officielle de Portsentry : https://sourceforge.net/projects/sentrytools/
- Documentation Proxmox sur les conteneurs LXC : https://pve.proxmox.com/wiki/Linux_Container

## 11.2 Glossaire

- **Fail2ban** : Outil de prévention d'intrusion qui protège les serveurs contre les attaques par force brute
- **Portsentry** : Outil de détection et de blocage des scans de ports
- **LXC** : Linux Containers, technologie de virtualisation au niveau du système d'exploitation
- **Proxmox** : Plateforme de virtualisation open source basée sur KVM et LXC
- **Force brute** : Technique consistant à essayer toutes les combinaisons possibles pour trouver un mot de passe
- **Scan de ports** : Technique utilisée pour découvrir les ports ouverts sur un système

---

Documentation réalisée par Ahmed Koucha - GSB - Mai 2025