ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

EMSI

Rapport de Projet

Mise en place d'un IPS avec Fail2Ban sur CentOS 9

Réalisé par :
Name1 NOM1
Name2 NOM2

Année Universitaire 2024-2025

Table des matières

Chapitre 1

Introduction

La sécurité des serveurs Linux est un enjeu critique dans l'administration système moderne. Les attaques par force brute (brute-force) constituent l'une des menaces les plus courantes, visant à deviner les mots de passe des services exposés comme SSH.

Ce projet a pour objectif de concevoir, configurer et valider un système de prévention d'intrusions (IPS) automatisé utilisant **Fail2Ban** sur une distribution **CentOS Stream 9**.

Le système sera capable de :

— Surveiller les journaux (logs) du système en temps réel.

— Détecter les tentatives de connexion échouées.

— Bannir dynamiquement les adresses IP malveillantes via le pare-feu (Firewalld).

— Protéger à la fois le service SSH et une application personnalisée simulée.

Chapitre 2

Architecture et Prérequis

2.1 Environnement Technique

— **Système d'exploitation :** CentOS Stream 9
— **Logiciel de sécurité :** Fail2Ban v1.x
— **Pare-feu :** Firewalld (intégré par défaut dans CentOS)
— **Services surveillés :** Daemon SSH (sshd) et une application métier personnalisée.

2.2 Préparation du Système

Avant l'installation, le système a été mis à jour et le dépôt EPEL (Extra Packages for Enterprise Linux) a été ajouté, car Fail2Ban n'est pas inclus dans les dépôts par défaut.

```
sudo dnf install epel-release -y
sudo dnf install fail2ban fail2ban-firewalld -y
sudo systemctl enable --now fail2ban
```

Listing 2.1 – Installation des paquets nécessaires

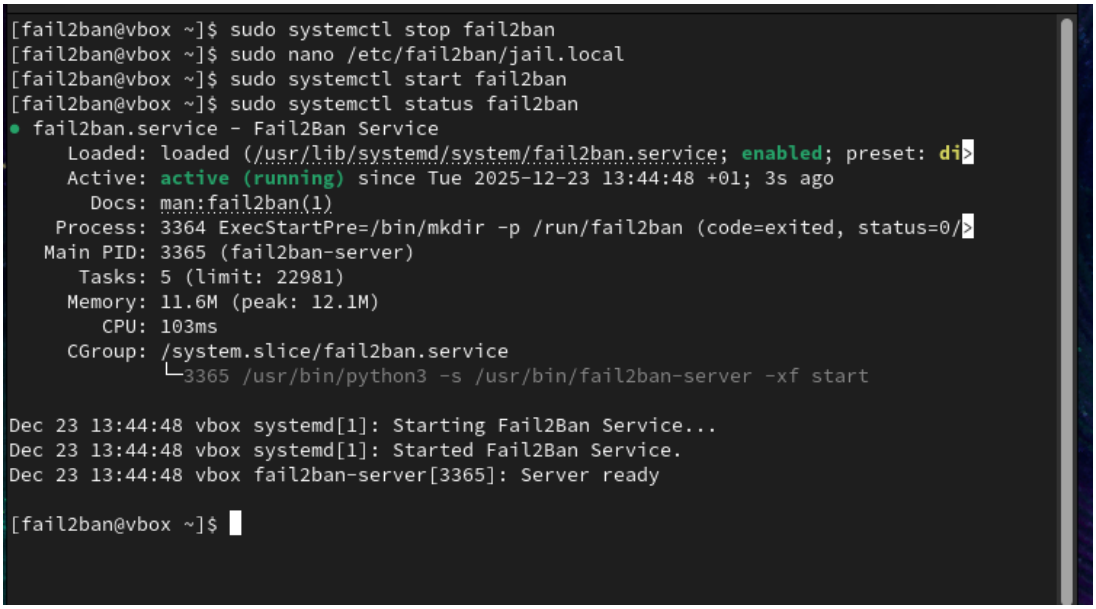

FIGURE 2.1 – Vérification de l'état du service Fail2Ban après installation.

Chapitre 3

Mise en Œuvre

3.1 Configuration de la Protection SSH

Pour protéger le service SSH, nous avons créé un fichier de configuration locale `/etc/fail2ban/jail.`
Une particularité de CentOS 9 est l'utilisation du backend `systemd` pour la lecture des logs SSH.

```
[DEFAULT]
bantime = 1h
maxretry = 3
findtime = 10m
ignoreip = 127.0.0.1/8 ::1

[sshd]
enabled = true
backend = systemd
mode = normal
```

Listing 3.1 – Configuration jail.local (Extrait SSH)

3.2 Création d'une Prison (Jail) Personnalisée

Afin de démontrer la flexibilité de Fail2Ban, nous avons simulé une application métier.

3.2.1 1. Création du fichier de log simulé

Nous avons créé un fichier vide qui recevra les logs de notre fausse application.

```
sudo touch /var/log/custom_app.log
sudo chmod 644 /var/log/custom_app.log
```

3.2.2 2. Création du Filtre (Regex)

Le filtre définit ce qu'est une "attaque". Nous avons configuré une expression régulière (Regex) dans `/etc/fail2ban/filter.d/custom-app.conf` pour détecter la chaîne "Failed login".

```
[Definition]
failregex = ^.*Failed login from <HOST>.*$
```

```
ignoreregex =
```

Listing 3.2 – Filtre personnalisé (custom-app.conf)

3.2.3 3. Activation de la Prison

Nous avons ajouté la configuration suivante à la fin de `jail.local` pour lier le filtre au fichier de log.

```
[custom-app]
enabled = true
filter = custom-app
logpath = /var/log/custom_app.log
maxretry = 3
backend = auto
```

Listing 3.3 – Ajout dans jail.local

Chapitre 4

Tests et Validation

Pour valider le fonctionnement, nous avons développé un script Bash simulant une attaque par force brute.

4.1 Script de Simulation d'Attaque

Ce script injecte 5 tentatives de connexion échouées provenant de l'IP fictive `192.0.2.100`.

```
#!/bin/bash
LOG_FILE="/var/log/custom_app.log"
ATTACKER_IP="192.0.2.100"

echo "Simulation d'attaque depuis $ATTACKER_IP..."
for i in {1..5}
do
   echo "$(date) Failed login from $ATTACKER_IP" | sudo tee -a $LOG_FILE
   sleep 1
done
```

Listing 4.1 – Script simulate_attack.sh

4.2 Résultats du Test

Après l'exécution du script, nous avons vérifié que Fail2Ban a bien détecté l'attaque et banni l'IP.

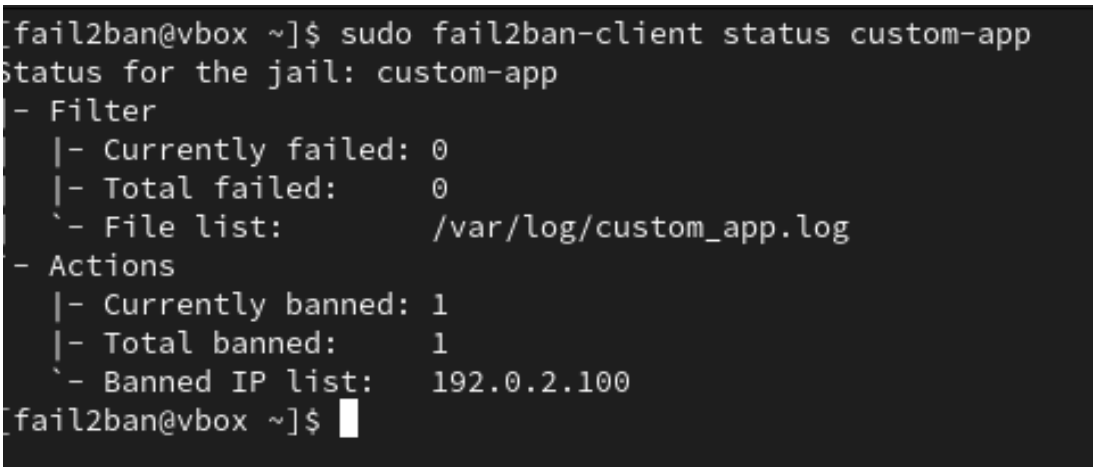

FIGURE 4.1 – Preuve que l'IP 192.0.2.100 a été bannie par Fail2Ban.

Enfin, nous avons vérifié que la règle a été propagée au niveau du pare-feu Firewalld.

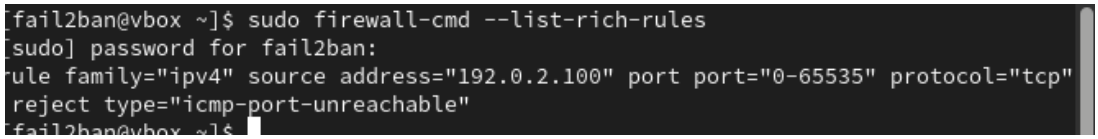

FIGURE 4.2 – Vérification de la règle Firewalld (Rich Rule).

Chapitre 5

Conclusion

Ce projet a permis de mettre en place une solution de sécurité robuste et automatisée sur CentOS 9. Grâce à Fail2Ban, le serveur est désormais capable de :

1. Détecter proactivement les attaques par force brute sur SSH.
2. Surveiller des applications personnalisées via des filtres Regex.
3. Bloquer instantanément le trafic malveillant via Firewalld.

La validation par simulation d'attaque a confirmé l'efficacité des prisons configurées. Ce système constitue une première ligne de défense essentielle pour tout serveur exposé sur Internet.