Sécurisation Automatisée avec Fail2Ban

Mise en place d'un IPS et filtres personnalisés sur CentOS Stream 9

Présenté par : **Name1** & **Name2**

Contexte : Projet d'Administration Système & Sécurité

Introduction et Problématique : Les Menaces Persistantes

La Menace Brute-Force

Les serveurs Linux exposés sur Internet sont la cible de tentatives de connexion automatisées (Brute-Force) visant à deviner les identifiants et mots de passe, en particulier via SSH.

Les Risques Encourus

- Saturation des ressources système (CPU, bande passante).
- Accès non autorisé au système (compromission).
- Exfiltration ou destruction de données sensibles.

La Solution : IPS Automatisé

Un **Système de Prévention d'Intrusions (IPS)** capable d'analyser les logs en temps réel et de bloquer dynamiquement les adresses IP malveillantes avant la compromission.

Objectifs Techniques du Projet

Déploiement Initial

Installer et configurer l'outil **Fail2Ban** sur l'environnement CentOS Stream 9.

Sécurisation SSH

Mise en place immédiate de la protection du service SSH (port 22) contre les tentatives d'épuisement de mots de passe.

Prison Personnalisée (Jail)

Développement d'un filtre Regex spécifique pour protéger une application métier non standard.

Intégration Firewalld

Assurer le bannissement efficace en utilisant les fonctionnalités avancées de **Firewalld** (Rich Rules).

Validation

Simuler une attaque pour confirmer l'efficacité et la réactivité du système IPS configuré.

Architecture de Défense et Flux d'Analyse

OS Hôte

CentOS Stream 9. Plateforme stable servant de base au déploiement du service.

Moteur d'Analyse

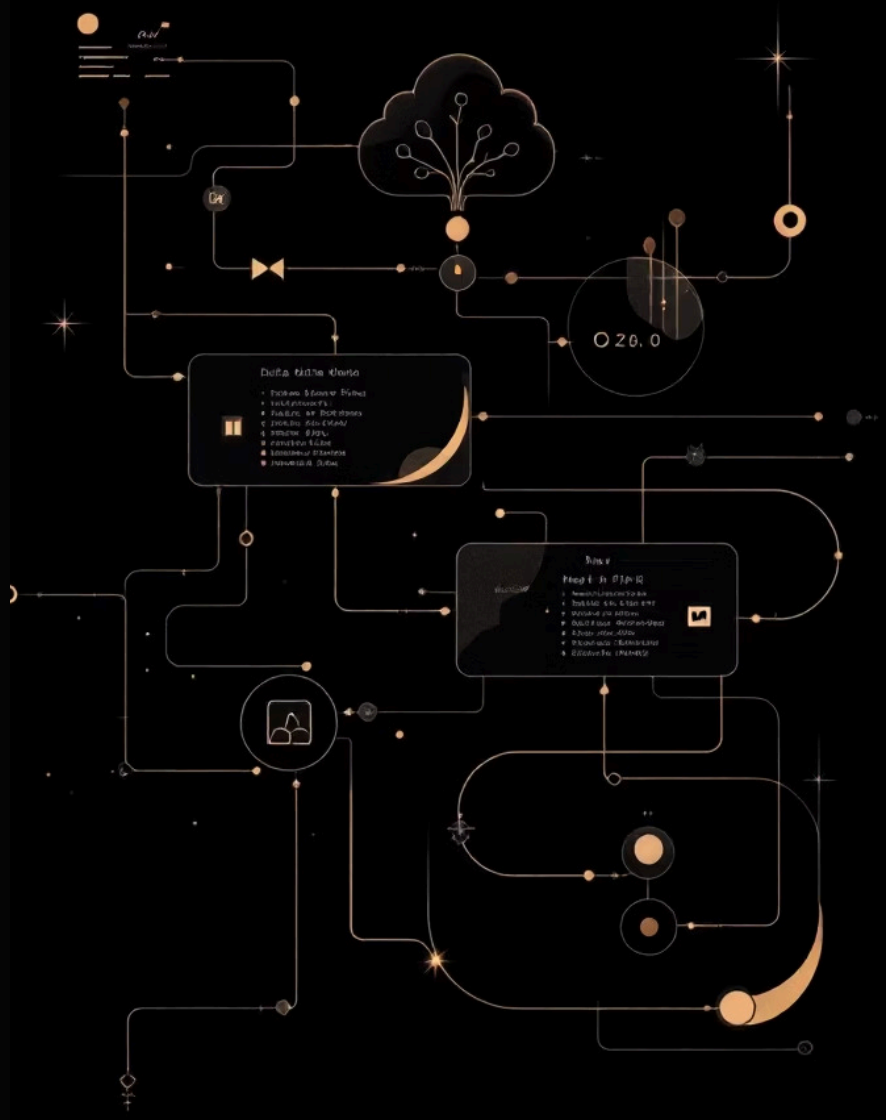
Fail2Ban. Scanne les logs systèmes en continu à l'aide de filtres (expressions régulières).

Outil d'Action

Firewalld. Exécute l'action de bannissement (Reject/Drop) via les "Rich Rules" pour une gestion granulaire du trafic.

Le Workflow de Défense

Tentative échouée → Log écrit → Filtre Regex activé → Action de Bannissement (Reject IP).



Phase 1 : Préparation et Installation du Système de Base

Ajout du Dépôt EPEL

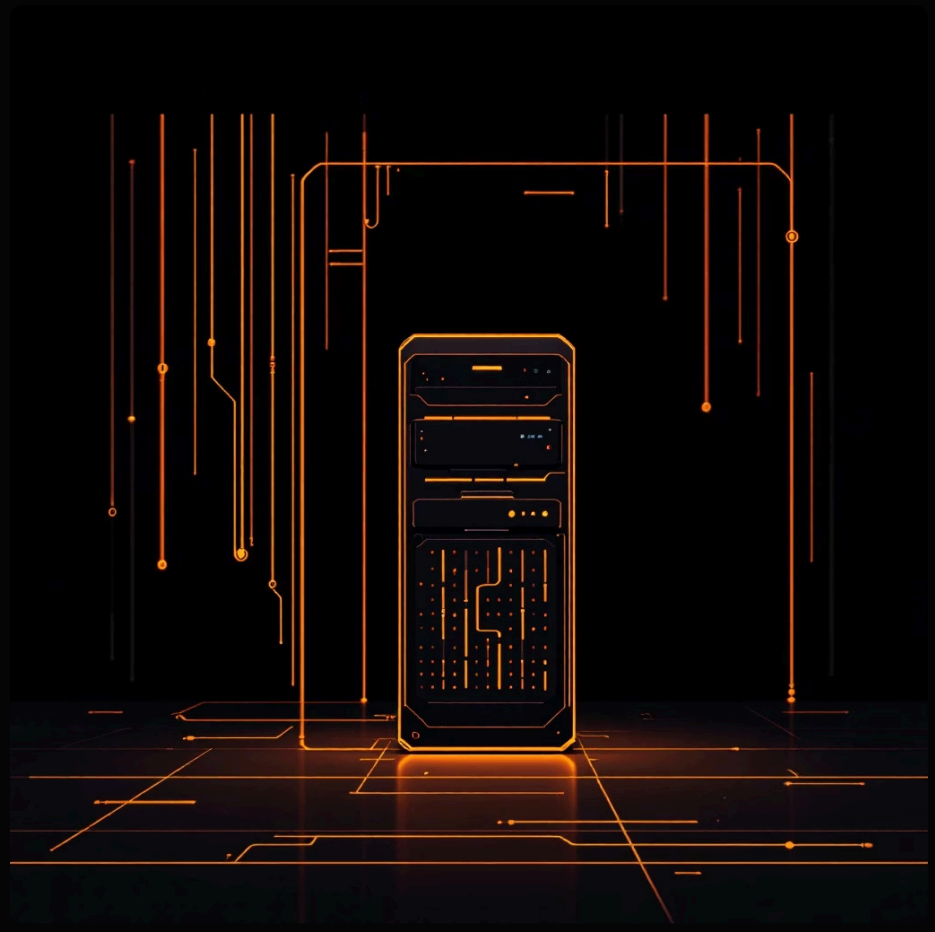
Fail2Ban ne se trouve pas dans les dépôts par défaut (AppStream). L'ajout du dépôt EPEL est indispensable pour l'installation des paquets nécessaires.

Installation des Paquets

Installation des dépendances clés : `fail2ban` et le module d'intégration au pare-feu `fail2ban-firewalld`.

Activation du Service

Démarrage et configuration pour un lancement automatique au boot via `systemctl enable fail2ban`.



Phase 2 : Sécurisation du Service SSH par Défaut

La protection SSH est la priorité. Elle est gérée via le fichier de configuration `/etc/fail2ban/jail.local`.

Backend Spécifique CentOS 9

Sur les systèmes modernes utilisant **Systemd** pour la journalisation (CentOS 9), il est critique de spécifier `backend = systemd` pour une lecture correcte des logs structurés (Journald).

Politique de Bannissement

- **maxretry :** 3 échecs tolérés.
- **findtime :** 10 minutes (période d'observation).
- **bantime :** 3600 secondes (1 heure de bannissement effectif).

Jail SSH

`[sshd]` activé pour cibler les logs d'authentification SSH. L'action par défaut utilise Firewalld pour la règle de rejet.

Phase 3 : Création d'une Prison Personnalisée (Custom Jail)

Pour étendre la protection au-delà des services standard, nous définissons un filtre Regex et une prison dédiés à une application métier simulée.

1. Définition du Log et du Filtre

- **Log Cible :** `/var/log/custom_app.log`
- **Filtre (Regex) :** Création de `custom-app.conf` dans `filter.d/`.

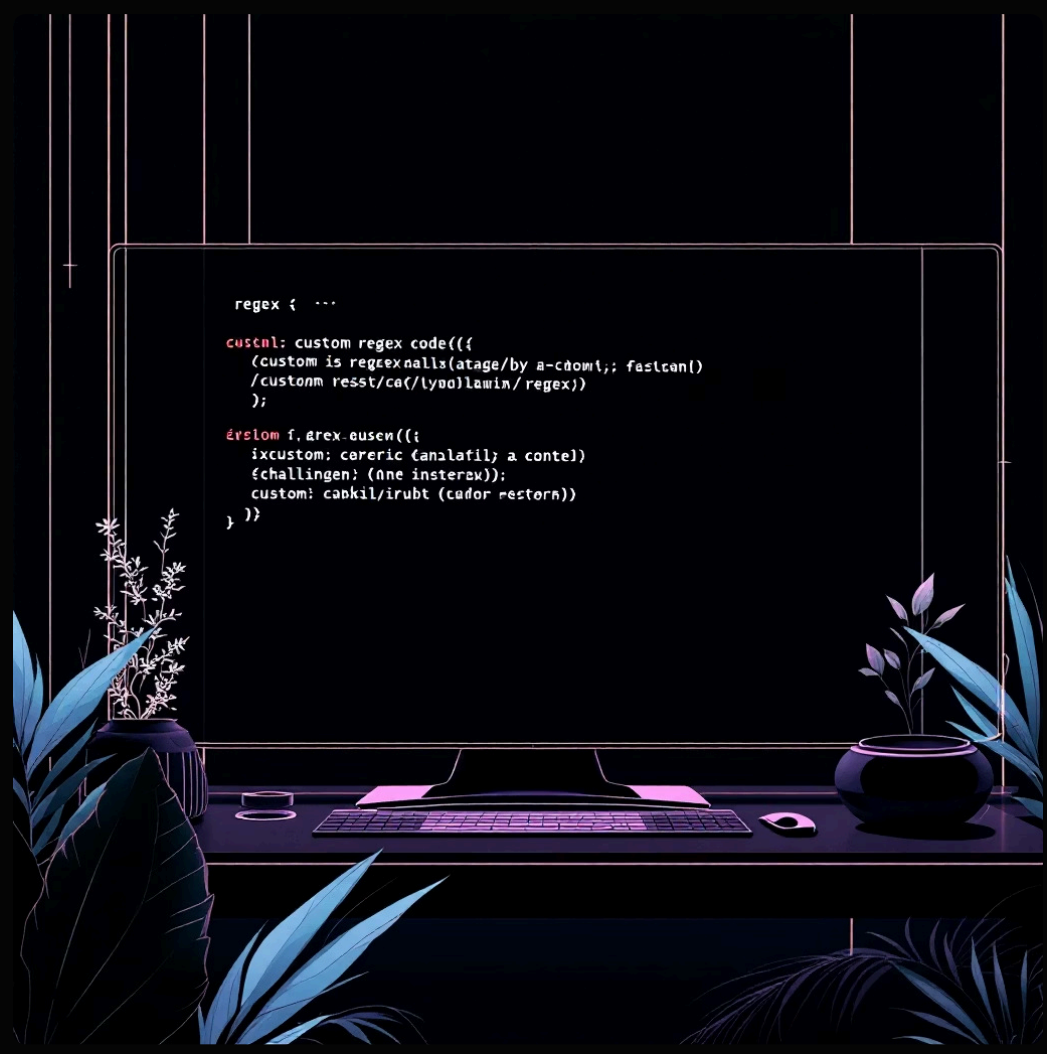
> Le Regex doit capturer l'IP hôte dans une tentative d'échec :
>
> ```
> ^.*Failed login from <HOST>.*$
> ```

Le balisage `<HOST>` indique à Fail2Ban où extraire l'adresse IP de l'attaquant.

2. Activation de la Jail

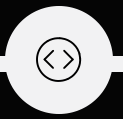
Ajout de la section suivante dans `jail.local` pour lier le filtre, le log et la politique de bannissement :

```
[custom-app]
enabled = true
logpath = /var/log/custom_app.log
filter = custom-app
maxretry = 3
bantime = 600
```



Stratégie de Test et Simulation d'Attaque

La validation de l'IPS nécessite de reproduire les conditions d'une attaque Brute-Force pour s'assurer du déclenchement automatique du bannissement.



Script de Simulation (Bash)

Développement d'un script `simulate_attack.sh` injectant des lignes de log factices directement dans `/var/log/custom_app.log`.

Injection d'IP Cible

L'adresse IP fictive **192.0.2.100** est utilisée pour simuler l'attaquant. Elle sera l'objet du bannissement.

Déclenchement du Ban

Le script injecte **5 lignes de log** d'échecs, dépassant le seuil de tolérance (`maxretry = 3`).

Observation Temps Réel

Observation des logs Fail2Ban pour confirmer la détection immédiate et l'exécution de l'action Firewalld.

Résultats et Preuves de Validation

1. Validation Fail2Ban-Client

La commande de statut confirme que l'IP malveillante a été identifiée et mise en liste noire par le moteur d'analyse.

```
# fail2ban-client status custom-app
Status for jail: custom-app
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     5
|  `- File list:        /var/log/custom_app.log
`- Actions
   |- Currently banned: 1
   |- Total banned:     1
   `- Banned IP list:   192.0.2.100
```

2. Preuve au Niveau du Pare-feu

L'intégration Firewalld est confirmée par l'ajout dynamique d'une "Rich Rule" de rejet (reject) ciblant l'adresse IP de l'attaquant.

```
# firewall-cmd --list-rich-rules
rule family="ipv4" source address="192.0.2.100"
reject type="icmp-port-unreachable"
```

Conclusion : Un Système de Défense Proactif et Évolutif

Défense Proactive

Déploiement réussi d'un IPS qui agit de manière autonome, réduisant la surface d'attaque sans nécessiter d'intervention humaine continue.

Maîtrise de l'OS

Validation de l'intégration spécifique à CentOS 9 (backend systemd) et de l'utilisation optimale de Firewalld.

Flexibilité des Filtres

Démonstration de la capacité à créer des filtres Regex personnalisés pour sécuriser toute application générant des logs d'authentification.

Robustesse et Production

Le système configuré est stable, validé et prêt à être déployé en environnement de production pour assurer une sécurité périmétrique renforcée.

La cybersécurité repose sur l'automatisation de la résilience.