|  |  |
| --- | --- |
|  | **Fail2ban Référence rapide** |

Fail2ban est un cadritiel (*framework*) de **prévention contre les intrusions**, écrit en Python.   
Il fonctionne sur les systèmes Linux possédant une **interface de contrôle des paquets** (tel que *TCP Wrapper*) ou un **pare-feu** (tel que *Netfilter*).

# Présentation

Fail2ban est une application qui **analyse les journaux de divers services** (SSH, Apache, FTP, …) en **cherchant des correspondances** **entre** **des motifs définis** **dans ses filtres** et les **entrées des journaux**.   
Lorsqu'une **correspondance est trouvée** une ou plusieurs **actions sont exécutées**.

Typiquement, fail2ban **cherche des tentatives répétées de connexions infructueuses** **dans les fichiers journaux** et procède à un **bannissement en ajoutant une règle au pare-feu iptables ou nftables** afin de bannir l'adresse IP de la source.

Dans la catégorie des **outils basés sur les journaux**, fail2ban est considéré comme le **meilleur bloqueur de force brute** basé sur les journaux.

Fondamentalement, comme tout autre bloqueur de force brute basé sur les journaux, fail2ban surveillera les fichiers journaux du système et lorsque certains événements configurés se produiront, ils déclencheront fail2ban pour bloquer l'hôte incriminé.

Voici les **principales fonctionnalités de fail2ban** …

* Il **s’exécute en tant que démon** (pas de délai pour effectuer des actions comme dans les outils basés sur cron) ;
* Il peut **utiliser différentes méthodes pour bloquer l'attaque** …  
  **iptables**  
  C'est la valeur par défaut, et sera certainement le meilleur choix pour la plupart des utilisateurs  
  **TCP Wrappers (/etc/hosts.deny)**Cela peut être particulièrement utile si on exécute un VPS qui n'a pas accès aux règles iptables  
  **Toute autre méthode** que l’on pourrait avoir besoin d'implémenter dans la configuration de son pare-feu   
  (il faudra alors définir soi-même les règles dans ce cas ;
* Il peut **gérer plusieurs services** …  
  sshd (par défaut), Apache2, vsFTPd/proftpd, … ;
* Il peut **envoyer des notifications par courriel** ;
* Il peut **interdire les adresses IP pour une durée limitée** ou **bannir définitivement les hôtes**.

# Installation

Le paquet fail2ban est disponible dans les dépôts APT …  
**>> sudo apt update  
>> sudo apt upgrade -y  
>> sudo apt install -y fail2ban**

Une fois installé, **Fail2ban démarrera automatiquement**.

S’il n’est pas démarré, on peut lancer le service fail2ban …  
**>> sudo** **systemctl start fail2ban.service**

**Remarque** …  
On peut également démarrer le service à l’aide …  
**>> sudo fail2ban-client -x start  
Server ready**ou  
**2021-08-20 12:00:11,805 fail2ban [7155]: ERROR Server already running**

On peut demander un **démarrage automatique** …  
**>> sudo systemctl enable fail2ban.service**

Et enfin de **contrôler si Fail2ban est bien installé** … **>> sudo systemctl status fail2ban.service  
● fail2ban.service - Fail2Ban Service  
 Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)  
 Active: active (running) since Fri 2021-08-20 12:02:00 EDT; 2s ago  
 Docs: man:fail2ban(1)  
 Process: 7185 ExecStartPre=/bin/mkdir -p /run/fail2ban (code=exited, status=0/SUCCESS)  
 Main PID: 7163 (fail2ban-server)  
 Tasks: 0 (limit: 2303)  
 Memory: 0B  
 CPU: 100ms  
 CGroup: /system.slice/fail2ban.service  
 ‣ 7163 /usr/bin/python3 /usr/bin/fail2ban-server --async -b -s /var/run/fail2>**

Si la réponse comporte du vert et les mots **active** (*running*) sur la ligne débutant par Active …  
le service est installé et actif.  
Si tel n’est pas le cas, on peut ensuite lancer la commande fail2ban-client afin de démarrer le service.

# Configuration

Les **fichiers de configuration** se trouve dans le **répertoire** **/etc/fail2ban.conf**.

Il est **vivement déconseillé de modifier les fichiers** de configuration **/etc/fail2ban/fail2ban.conf** **et /etc/fail2ban/jail.conf**.   
Ces **fichiers peuvent être écrasés par une mise à jour**.   
Ces fichiers devraient seulement **servir uniquement de référence et de documentation**.

Ces fichiers contiennent les configurations de base qu'on peut **compléter au moyen d'un ou plusieurs fichiers sauvegardés** dans le **répertoire /etc/fail2ban/jail.d**.

Fail2ban **active par défaut la protection contre les attaques par force brute SSH**.   
Le fichier de configuration contient chaque paramètre abondamment commenté.

## Généralités

Fail2ban ne **doit pas être considéré comme un outil de sécurisation absolu d'un service**.   
Ses **objectifs sont d'éviter de surcharger les journaux** (*logs*) du système avec des **milliers de tentatives de connexion** et de **limiter la** **portée des attaques répétées** **provenant d'un même hôte**.

Un serveur avec un accès SSH sur le port standard, par exemple, recevra très rapidement des centaines, voire des milliers de tentatives de connexions provenant de différents hôtes depuis Internet.   
Ce sont généralement des attaques par force brute lancées par des robots.

Fail2ban en analysant les journaux (*logs*) **permet de bannir les adresses IP au bout d'un certain nombre de tentatives** ce qui **limitera le remplissage des journaux** et **l'utilisation de la bande passante**.  
Ceci va également rendre les attaques par force brute ou par dictionnaire beaucoup plus difficiles.

Toutefois, ce n'est pas une **sécurité absolue contre ce type d'attaque**.

**Remarque** …  
Cela **n'améliore en rien la sécurité du service SSH**.   
Si l'accès SSH n'est pas suffisamment sécurisé (mot de passe faible par exemple) fail2ban n'empêchera pas un attaquant d'arriver à ses fins.

## Paramètres par défaut

Les paramètres par défaut se retrouvent dans le fichier /etc/fail2ban/jail.conf

* La **durée de bannissement** d'une adresse IP est définie par la **directive bantime** avec une **valeur en secondes**.   
  La valeur par défaut est de 600 s (soit 10 minutes), ce qui est beaucoup trop court.   
  Il est plus réaliste d'avoir des durées de bannissement d'une ou plusieurs heures ou de plusieurs jours.
* Une autre importante **directive** est **findtime** qui **définit en secondes** le **temps depuis lequel une anomalie est recherchée** **dans les journaux**.   
  Il ne faut pas mettre une valeur trop élevée (1 heure suffit) sans quoi la quantité de journaux à analyser pourrait devenir très importante et donc avoir un impact sur les performances.
* La **directive ignoreip** permet de définir la **liste des adresses IP à ignorer**.   
  Il est **utile d'y mettre sa propre adresse IP** afin de ne pas risquer de se faire bannir.

Il faut créer le **fichier dans le répertoire /etc/fail2ban/jail.d/custom.conf** (ou un autre nom de son choix) contenant …  
**[DEFAULT]  
ignoreip = 127.0.0.1 192.168.1.100  
findtime = 3600** # 10 minutes  
**bantime = 86400** # 24 heures **maxretry = 3**

Description de quelques-uns des **paramètres importants de la section principale [DEFAUT]** …

* La **directive ignoreip** permet de préciser une **liste séparée par des espaces des adresses IP** à ignorer par fail2ban  
  il est possible de **spécifier une étendue d'adresses**.   
  Par exemple, 192.168.1.0/24 désigne les adresses de 192.168.1.0 à 192.168.1.25 ;
* Si la **directive bantime** est définie sur une **valeur négative** (bantime = -1), l'adresse IP ne sera jamais débloquée automatiquement (interdiction permanente) ;
* La **directive maxfailures ou maxretry** précise le **nombre d'échecs avant que l'adresse IP ne soit bannie**.   
  La valeur par défaut est 5. Beaucoup d’administrateurs système réduisent cette valeur à 3.

## Configuration des services actifs

Pour spécifier à fail2ban quels **services il doit surveiller**, il faut **activer les cellules** (*jails*) **correspondantes**.

En consultant le **fichier /etc/fail2ban/jail.conf**, on retrouve dans la **partie jail des blocs du type** …  
**[sshd]  
port = ssh  
logpath = %(sshd\_log)s  
backend = %(sshd\_backend)s**

Quelques explications …

* **port**  
  Ports à bloquer au moyen des règles iptables ;
* **logpath**  
  Emplacement des fichiers de log à surveiller ;
* **backend**  
  Moteur de surveillance des logs.

Les valeurs représentées ainsi **%(sshd\_logs)s sont des variables** qui sont définies dans d'autres fichiers de configuration …  
notamment /**etc/fail2ban/paths\_common.conf** et **/etc/fail2ban /paths\_debian.conf**.

Pour activer la **surveillance des connexion SSH**, il suffit d'**ajouter** **dans le fichier /etc/fail2ban/jail.d/custom.conf** …  
**[sshd]  
enabled = true**

**Remarque** …  
Il suffit de procède de même pour les autres services.  
Il est bien entendu **inutile d'activer des cellules** (*jails*) **pour des services qui ne sont pas utilisés** sur le serveur.

Si on a besoin de **spécifier un autre port** comme lorsque SSH n'est pas en écoute sur un port standard, un fichier de journal (*log*) particulier ou un nombre de tentatives différent de la valeur par défaut, il faut le **préciser dans le fichier /etc/fail2ban/jail.d/custom.conf** …  
**[sshd]  
enabled = true  
port = 2222  
logpath = /var/log/auth.log  
maxretry = 5**

D'une manière générale, il est important d'effectuer des tests pour vérifier le bon fonctionnement de fail2ban.

À chaque modification d’n fichier de configuration, on doit relancer la configuration …  
**>> sudo** **systemctl restart fail2ban.service**

Si on veut vérifier si les cellules (*jails*) ont été correctement lancées …  
**>> sudo** **fail2ban-client status**  
**Status  
|- Number of jail: 3  
`- Jail list: apache, proftpd, sshd**

Les cellules (*jails*) peuvent être contrôlées séparément avec les mots clés start, stop et status …  
**>> sudo** **fail2ban-client stop sshd  
Jail stopped**

## Vérification du bon fonctionnement

Pour vérifier le bon fonctionnement de fail2ban on peut essayer de **s’identifier plusieurs fois en saisissant un mauvais mot de passe**.   
Si Fail2ban fonctionne, l'accès au serveur devrait être interdit au bout d'un certain nombre d'essais   
(précisés dans la configuration à la **directive maxretry**) à condition de **ne pas avoir mis son adresse IP dans la directive ignoreip**.

Remarque …  
il est préférable de **régler la valeur de la** **directive bantime** sur un temps assez court si on fait ce genre d’essais **afin de pouvoir se** **reconnecter au serveur**.

Côté serveur, il est également possible de surveiller ce qui se passe …  
**>> sudo** **fail2ban-client status sshd  
Status for the jail: sshd  
|- Filter  
| |- Currently failed: 23  
| |- Total failed: 49920  
| `- File list: /var/log/auth.log  
`- Actions  
 |- Currently banned: 6  
 |- Total banned: 7003  
 `- Banned IP list: 50.76.88.187 193.224.98.111 46.242.131.56 49.88.112.113 76.116.113.183 179.53.239.248**

Cette commande retournera le **statut de la cellule** (*jail*) sshd avec le **nombre de tentatives échouées** et la **liste des adresses IP** **bloquées**.

Il est aussi possible d’examiner les journaux (*logs*) de fail2ban pour voir les actions effectuées …  
**>> sudo tail -f /var/log/fail2ban.log**

Document adapté de la page …  
**https://doc.ubuntu-fr.org/fail2ban**

  