

Fail2ban pour Joomla sur Debian10 (Turnkey16) derrière un reverse proxy

Difficulté



Moyen

HAProxy

- Régler pour obtenir les IPs de connexion tout en étant derrière le proxy: [HAProxy](#)
- Dans Joomla: Système ⇒ Configuration ⇒ Serveur: aller tout en bas à Paramètres du proxy et activer Behind Load Balancer



<https://www.triumvirat.org/posts/opnsense/opnsense-fail2ban-api>

<https://www.ncatron.org/opnsense-api.html/>

OPNSense

Privilèges de l'utilisateur: "Diagnostics: PF Table IP addresses"

Fail2ban de base

source: <https://linuxize.com/post/install-configure-fail2ban-on-debian-10/>

- fail2ban est configuré d'origine sur ct Turnkey.

```
#systemctl status fail2ban
```

le confirme

- installer whois pour des infos plus détaillées sur les IPs bloquées

```
# apt-get install whois
```

- copier le fichier de configuration d'origine qui ne doit pas être modifié

```
# cp /etc/fail2ban/jail.{conf,local}
```

- paramétrer dans cette copie

```
# nano /etc/fail2ban/jail.{conf,local}
...
ignoreip = 127.0.0.1/8 IPs.reseau.lan.local/24 si besoin
...
bantime = 60m
```

```
...  
findtime = 10m  
...  
maxretry = 5  
...  
destemail = admin@domain.tld  
...  
action = %(action_mw)s  
...
```

- penser à activer les jails souhaitées au passage

```
...  
enabled = true  
...
```

```
# systemctl restart fail2ban
```

Pour Joomla

Malheureusement les jails de fail2ban ne suffisent pas car ce n'est pas Apache qui logge pas les tentatives d'identification mais Joomla!

Sources: <https://www.joomla-security.de/server/joomla-login-mit-fail2ban-schuetzen.html> et <https://www.andrehotzler.de/de/blog/technology/62-joomla-login-mit-fail2ban-schuetzen.html>

- créer le filtre:

```
# nano /etc/fail2ban/filter.d/joomla-login-errors.conf  
[Definition]  
failregex = ^.*INFO  
<HOST>.*joomlafailure.*(Benutzername|Username|utilisateur).*
```



Mettre toutes les traductions des langues utilisées dans le site

Fail2ban

- créer la jail:

```
# nano /etc/fail2ban/jail.d/joomla-login-errors.conf  
[joomla-login-errors]  
enabled = true
```

```
filter = joomla-login-errors
port   = http,https
logpath = /var/www/joomla/administrator/logs/error.php
```

- # systemctl restart fail2ban

Restreindre l'accès au répertoire d'administration

- Dans la config d'Apache:

```
# nano /etc/apache2/sites-available/joomla.conf

et ajouter:
<Directory /var/www/joomla/administrator>
    Order Deny,Allow
    Deny from all
    Allow from ip1.ou.bien.réseau1/24 ip2.ou.bien.réseau.2/24
</Directory>
```

- # systemctl restart apache2

Communiquer avec OPNSense

Il faut créer un petit script:

```
# nano fail2ban-IP.sh
```

```
#!/bin/sh
KEY="la_clé"
SECRET="le_mot_de_passe"
FWIP="ip.de.la.OPNSense"
FWPORT="n°_de_port"
ALIAS="fail2ban_CT_Joomla"
if [ $1 = "ban" ]; then
    TODO="add"
elif [ $1 = "unban" ]; then
    TODO="delete"
fi
curl -X POST -d '{"address":"'${2}'"' -H "Content-Type: application/json" -k
-u $KEY:$SECRET https://$FWIP:$FWPORT/api/firewall/alias_util/$TODO/$ALIAS
```

- La clé et le mot de passe sont les identifiants API de l'utilisateur créé à cet effet
- le n° de port est celui utilisé pour la connexion https à l'interface web
- l'alias est celui utilisé pour la règle de blocage du firewall

```
# chmod 600 fail2ban-IP.sh  
# chmod +x fail2ban-IP.sh
```

Pour tester le script:

```
# ./fail2ban-IP.sh ban 192.168.1.1  
# ./fail2ban-IP.sh unban 192.168.1.1
```

et pour finir:

```
# systemctl restart fail2ban  
# systemctl status fail2ban
```

```
# fail2ban-client set joomla-login-errors banip 192.168.1.1  
# fail2ban-client set joomla-login-errors unbanip 192.168.1.1
```

From:
<https://wiki.guedel.eu/> - **Wiki-Guedel**

Permanent link:
https://wiki.guedel.eu/doku.php?id=welcome:linux_usually:secure_joomla&rev=1639675049

Last update: **2021/12/16 17:17**

