# Faire un pare-feu d'application web (waf) avec apache



vous l'aurez compris dans le titre nous allons installer le mode `security2` sur apache avec les règles OWASP CRS, pour obtenir un WAF.
Un pare-feu d'application Web (WAF) est un pare-feu applicatif pour les applications HTTP(S).
Il applique un ensemble de règles à une conversation HTTP(S).
Généralement, ces règles couvrent les attaques courantes telles que le cross-site scripting (XSS), l'injection SQL, etc.

Bien que les proxies protègent généralement les clients, les WAF protègent les serveurs.
Un WAF est déployé pour protéger une application Web spécifique ou un ensemble d'applications Web.
Un WAF peut être considéré comme un proxy inverse.
Les WAF peuvent se présenter sous la forme d'une appliance, d'un plugin de serveur ou d'un filtre et peuvent être personnalisés pour une application.
L'effort pour effectuer cette personnalisation peut être important et doit être maintenu quand l'application est modifiée.

## Mise en place des packages



A ce stade, il est important de disposer d'un serveur Apache installé

Nous allons installer le module security2 d'apache:
Sur Debian ou équivalent :

```
aptitude install libapache2-mod-security2
```

## configuration du modSecurity

Le répertoire qui contiendra les règles et les fichiers de configuration seront situés dans `/etc/modsecurity/`, dans ce répertoire un fichier `modsecurity.conf-recommended` contenant quelques règles mais aussi le paramètre qui permet d'activer le modSecurity.
Dans un premier temps, il faut renommer le fichier `modsecurity.conf-recommended` en `modsecurity.conf`.

```
cp /etc/modsecurity/modsecurity.conf-recommended
/etc/modsecurity/modsecurity.conf
```

Puis ont modifie le paramètre `SecRuleEngine` pour activer les règles.

```
vim /etc/modsecurity/modsecurity.conf
```

Puis modifier `SecRuleEngine` comme cela

```
SecRuleEngine On
```

Dans le fichier de configuration `security2.conf`, indiquer le chemin du fichier que nous venons de modifier.

```
vim /etc/apache2/mods-available/security2.conf
```

Puis modifier ou ajouter la ligne suivante :

```
<IfModule security2_module>
      ...
      IncludeOptional /etc/modsecurity/*.conf
      ...
</IfModule>
```

Puis activer le module:

```
# a2enmod security2
Considering dependency unique_id for security2:
Module unique_id already enabled
Enabling module security2.
To activate the new configuration, you need to run:
service apache2 restart
```

Penser à redémarrer ou recharger la configuration :

```
/etc/init.d/apache2 restart
```

ou

```
/etc/init.d/apache2 reload
```

## Mise en place des règles OWASP CRS

Le jeu de règles de base (CRS) ModSecurity d'OWASP est un ensemble de règles génériques de détection d'attaque à utiliser avec ModSecurity ou des pare-feu d'application Web compatibles.
Le CRS vise à protéger les applications Web d'un large éventail d'attaques, avec un minimum de fausses alertes.
Nous allons mettre en place cela avec GIT:

```
aptitude install git
```

Téléchargement des règles dans `/etc/modsecurity/` :

```
cd /etc/modsecurity/
git clone -b v3.0/master
https://github.com/SpiderLabs/owasp-modsecurity-crs.git
```

Après le téléchargement, copier le fichier `crs-setup.conf.example` en `crs-setup.conf`.
Prenez le temps de parcourir ce fichier et de personnaliser les paramètres pour votre environnement local, car vous risquez d'avoir des faux positifs.
Voir la section configuration OWASP CRS.
Renommer les règles `/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example` et règles `/RESPONSE-999-EXCLUSION-RULES-AFTER-CRS.conf.example` pour supprimer l'extension `.example`.
Cela vous permettra d'ajouter des exclusions.
Ajoutez les lignes suivantes à votre fichier `/etc/apache2/mods-available/security2.conf`

```
vim /etc/apache2/mods-available/security2.conf
```

Puis modifier ou ajouter la ligne suivante :

```
<IfModule security2_module>
      ...
      Include modsecurity.d/owasp-modsecurity-crs/crs-setup.conf
      Include modsecurity.d/owasp-modsecurity-crs/rules/*.conf
      ...
</IfModule>
```

Redémarrez le serveur apache2 et assurez-vous qu'il démarre sans erreurs. Assurez-vous que vos sites Web fonctionnent toujours très bien.



sur debian avec la version 2.4.10
J'ai eu au redémarrage un message de ce genre :

```
[FAIL] Restarting web server: apache2 failed!
[warn] The apache2 configtest failed. ... (warning).
Output of config test was:
AH00526: Syntax error on line 36 of /etc/modsecurity/owasp-
modsecurity-crs/rules/RESPONSE-950-DATA-LEAKAGES.conf:
Error parsing actions: Unknown action: \\
Action 'configtest' failed.
The Apache error log may have more information.
```

**Résolution :** Éditer le fichier `/etc/modsecurity/owasp-modsecurity-crs/rules/RESPONSE-950-DATA-LEAKAGES.conf` et à la ligne 36 vous trouverez cela :

```
t:none,\
```

Modifier en ajoutant un caractère ESPACE entre la `,` et `\`

```
t:none, \
```

Si vous ne souhaitez pas désactiver complètement `modsecurity`, utilisez la directive `SecRuleRemoveById` pour supprimer une règle ou une chaîne de règles particulière en spécifiant

son ID, dans votre “VirtualHost”:

```
<IfModule security2_module>
    SecRuleRemoveById 920420
    SecRuleRemoveById 949110
    ...
</IfModule>
```

From:
https://www.ksh-linux.info/ - **Know Sharing**

Permanent link:
**https://www.ksh-linux.info/systeme/apache/faire-un-firewall-d-application-web-waf-apache**

Last update: **20/02/2019 21:25**