# Introduction :

Mod\_security est un module d’apache (open source) qui joue le rôle d’un firewall de type applicatif ( anlyse au niveau 7 d la couche OSI), et dont l’objectif est de filtrer les requêtes en entrée destinées au serveur web, et les réponses données par ce dernier, mais aussi de loguer les transactions.

# Installation de mod\_security :

## Pré-requis :

Avant de lancer l’installation de mod\_security (comme module d’apache), il faut s’assurer d’une part, qu’on est en train de travailler sur une version d’apache égale ou supérieur à 2.0.x, et d’autre part il faudrait s’assurer aussi que le package mod\_unique\_id d’apache est installé, et enfin il faut s’assurer que les librairies suivantes sont installées :

* + - * libapr
			* libapr-util
			* libpcre
			* libxml2
			* liblua v5.1.4
			* libcurl v7.15.1 (ou version supérieur)

## Installation :

Les différentes étapes de l’installation de mod\_security sont :

* Mettre à l’arrêt le serveur apache (pas obligatiore !)
* Télécharger de la dernière version stable de mod\_security sous forme d’archive à partir du site suivant :

http://www.modsecurity.org/download/

* L’archive sera placé dans le répertoire /install\_local, et décompressé.
* Compiler et installer mod\_security
	+ ./configure
	+ make
	+ make install
* Copier le fichier /install\_local/modsecurity-apache\_2.6.3/modsecurity.conf-recommended, dans le dossier /usr/local/apache2/conf/extra sous le nom de modsecurity.conf.
* **Rajouter ces lignes au fichier httpd.conf :**

 **-LoadFile /usr/lib64/libxml2.so**

 **-LoadFile /usr/lib64/liblua-5.1.so**

 **-LoadModule security2\_module modules/mod\_security2.so**

* Redémarrer le serveur apache.

NB : En cas de problème de fonctionnement de mod security, (non prise en charge par apache) rajouter (--pcre switch) au paramètre de compilation de mod\_security. (soit ça marche, soit ça marche pas, ça a été testé en Dev …..

## Configuration de mod\_security :

Il va y avoir plusieurs configuration à faire pour bien faire fonctionner mod\_security, notamment en créant des fichiers de log, ou en téléchargeant un ensemble de règles déjà écrite, voir de procéder à des modifications au niveau du fichier : /usr/local/apache2/conf/extra/modsecurity.conf.

### Création des fichiers log de mod\_security :

Nous allons créer deux fichiers log qui seront utilisé par mod\_security : debug.log, et modsec\_audit.log (ces deux fichier seront créer automatiquement par mod\_security), ainsi qu’un dossier audit (il doit servir à stocker les audit de log) et tout cela dans le répertoire : /log/VMDEVSECU1/mod\_security.

### Téléchargement, et mise en place du core rules de mod\_security :

 A partir de cette URL : <http://sourceforge.net/projects/modsecurity/files/modsecurity-crs/> on va télécharger un ensemble de fichiers de configuration (contenant des règles de filtrages), ou de données, qui seront utilisée pour mettre en place une protection générique des applications web se trouvant sur apache.

 Le dossier télécharger (modsecurity-crs\_2.2.4.tar.gz), serra placer dans le répertoire /install\_local, pour être décompresser, une fois décompressé le dossier modsecurity-crs\_2.2.4 serra placé dans le répertoire /usr/local/apache2/conf/extra/crs

### Réglage à faire dans mod\_security :

 A ce stade de la configuration nous devons :

#### Activer des fichiers de core rules :

 Activer les fichiers core rules qu’on juge les plus pertinents et cela en créant des liens symbolique de ces derniers dans le répertoire /usr/local/apache2/conf/extra/crs/modsecurity-crs\_2.2.4/activated\_rules, et cela pour que les règles contenues dans ces derniers soit prises en compte par mod\_security.

 Jusqu'à maintenant on a activé tout les fichiers se trouvant dans le répertoire /usr/local/apache2/conf/extra/crs/modsecurity-crs\_2.2.4/base\_rules (le minimum de règles à activer pour un bon fonctionnement de mod security) et cela en utilisant le script suivant :

|  |
| --- |
| for f in `ls base\_rules/` ; do ln -s /usr/local/apache2/conf/crs/modsecurity-crs\_2.2.4/base\_rules/$f /usr/local/apache2/conf/extra/crs/modsecurity-crs\_2.2.4/activated\_rules/$f; done |

###  Après on duplique le fichier /usr/local/apache2/conf/extra/crs/modsecurity-crs\_2.2.4/modsecurity\_crs\_10\_config.conf.example, dans le répertoire /usr/local/apache2/conf/extra/crs/mod\_security-crs\_2.2.4 sous le nom de modsecurity\_crs\_10\_config.conf, avant de lui créer un lien symbolique dans le répertoire activated\_rules, (ce fichier est indispensable au bon fonctionnement d’autres fichiers de configuration, puisqu’ils peuvent partager les mêmes variables global).

#### Paramétrer le fichier modsecurity.conf :

 Les opérations suivantes doivent être faites dans le fichier /usr/local/apache2/conf/extra/modsecurity.conf.

* Remplacer la règle : **SecRuleEngine DetectionOnly**, Par : **SecRuleEngine on** (le mode DetectionOnly fera fonctionner mod-security comme un IDS, alors que le mode on le fera fonctionner comme un firewall applicatif).
* Remplacer la règle suivante : **SecDebugLog /opt/modsecurity/var/log/debug.log**Par : **SecDebugLog /log/VMDEVSECU1/modsecurity/debug.log**
* Remplacer la règle suivante : **SecAuditLog /var/log/modsec\_audit.log**Par : **SecAuditLog /log/VMDEVSECU1/modsecurity/modsec\_audit.log**
* Remplacer la règle suivante : **SecAuditLogStorageDir /opt/modsecurity/var/audit/** Par : **SecAuditLogStorageDir /log/VMDEVSECU1/modsecurity/audit/**
* Remplacer la règle : **SecAuditEngine RelevantOnly** Par : **SecAuditEngine off**(On peut se passer des informations fournit par le fichier modsec\_audit.log, ce qui nous permettra surtout d’économiser des ressources)

## Intégration dans la configuration d’apache :

Pour prendre en compte les règles du core-rules nous allons rajouter ces lignes à httpd.conf

|  |
| --- |
| Include conf/extra/modsecurity.conf<IfModule security2\_module> Include conf/crs/modsecurity-crs\_2.2.4/modsecurity\_crs\_10\_config.conf Include conf/crs/modsecurity-crs\_2.2.4/activated\_rules/\*.conf</IfModule> |

# Test :

Nous allons procéder à un test et ceci en naviguant dans le site web avec mod\_security activé pour voir si tout fonctionne normalement (il n’y a pas de page bloqué, pas de faux positif, vérifié que les fichiers log d’apache ou de mod\_security ne sont pas entrain de grossir anormalement).

On se rend compte que la navigation dans le site web créer de très nombreux faux positif qui sont tous en rapport avec la règle 950901 de mod\_security qui se trouve dans le fichier :modsecurity\_crs\_41\_sql\_injection\_attacks.conf, c’est pour cela il faudrait désactiver la partie SQL tautologie (en la commentant).