Voici une démonstration complète de l'utilisation des **règles dans `mod_security`**,

---

Démonstration Complète : `mod_security` dans Apache

1. Installer et Configurer `mod_security`

Installation sur une distribution basée sur Debian (Ubuntu) :

```
sudo apt update
sudo apt install libapache2-mod-security2
```

Activer le module `mod_security` dans Apache :

```
sudo a2enmod security2
sudo systemctl reload apache2
```

Vérifier que le module est chargé :

```
apache2ctl -M | grep security2
```

---

2. Configurer `mod_security`

Localisation des fichiers de configuration :

- **Fichier principal :** `/etc/modsecurity/modsecurity.conf`
- **Fichier de règles personnalisées :** `/etc/modsecurity/custom_rules.conf`

Activer le mode Détection uniquement (mode non bloquant) :

Dans le fichier `/etc/modsecurity/modsecurity.conf`, recherchez et modifiez la ligne suivante :

```
SecRuleEngine DetectionOnly
```

- `DetectionOnly` : ModSecurity journalise les événements mais ne bloque rien.
- `On` : Activer les règles pour bloquer les requêtes.

Inclure vos règles personnalisées :

Ajoutez cette ligne à la fin de `modsecurity.conf` pour inclure un fichier de règles spécifiques :

```
Include /etc/modsecurity/custom_rules.conf
```

---

3. Créer des Règles Personnalisées

Fichier des règles : `/etc/modsecurity/custom_rules.conf`

1. **Bloquer une Requête Contenant une Injection SQL :**

```
SecRule ARGS "select.+from" "id:1001,phase:2,deny,status:403,msg:'SQL
Injection Detected'"
```

2. **Bloquer les Téléchargements de Fichiers PHP :**

```
SecRule FILES_NAMES "\.php$" "id:1002,phase:2,deny,status:403,msg:'Blocked
PHP Upload'"
```

3. **Limiter la Taille des Requêtes POST :**

```
SecRequestBodyLimit 102400
SecRule REQUEST_BODY_LENGTH "@gt 102400"
"id:1003,phase:2,deny,status:413,msg:'Request Body Too Large'"
```

4. **Bloquer un User-Agent Spécifique :**

```
SecRule REQUEST_HEADERS:User-Agent "curl"
"id:1004,phase:1,deny,status:403,msg:'Blocked curl User-Agent'"
```

Activer et Tester les Règles

1. Rechargez Apache après avoir ajouté les règles :

```
sudo systemctl reload apache2
```

2. Testez avec `curl` :
   - Injection SQL :

```
curl "http://localhost?user=admin&query=select+*+from+users"
```

- Téléchargement de fichier PHP :

```
curl -F "file=@malicious.php" http://localhost/upload
```

- Taille de requête POST :

```
curl -X POST -d @largefile.txt http://localhost/upload
```

- User-Agent `curl` :

```
curl -A "curl" http://localhost
```

---

4. Analyser les Logs

Les événements détectés ou bloqués sont journalisés dans :

```
/var/log/apache2/modsec_audit.log
```

Exemple d'entrée dans les logs :

```
--f2c2b1-A--
[2024-12-05 14:30:22.123456] [id "1001"] [msg "SQL Injection Detected"]
[client 192.168.1.10] [uri "/index.php"] [args
"user=admin&query=select+*+from+users"]
Action: Intercepted (phase 2)
```

---

5. OWASP Core Rule Set (CRS)

Pour une protection avancée, vous pouvez activer l'ensemble de règles OWASP CRS. Cela ajoute des protections prêtes à l'emploi contre :

- Injections SQL.
- XSS (Cross-Site Scripting).
- CSRF (Cross-Site Request Forgery).
- Dépassements de taille de requête.

Installation des règles OWASP CRS :

1. Installez le CRS :

```
sudo apt install modsecurity-crs
```

2. Incluez les règles dans la configuration principale de `mod_security` : Ajoutez ces lignes dans `/etc/modsecurity/modsecurity.conf` :

```
IncludeOptional /usr/share/modsecurity-crs/*.conf
IncludeOptional /usr/share/modsecurity-crs/rules/*.conf
```

3. Rechargez Apache :

```
sudo systemctl reload apache2
```

---

6. Bonnes Pratiques

1. **Testez en mode `DetectionOnly` avant de bloquer des requêtes.**
2. **Analysez les logs pour affiner vos règles et réduire les faux positifs.**
3. **Donnez un ID unique à chaque règle pour faciliter le suivi.**
4. **Utilisez les règles OWASP CRS comme base et ajoutez des règles spécifiques à vos besoins.**

---

7. Résumé

- **Installation et configuration :** Vous installez `mod_security`, configurez son mode, et incluez des fichiers de règles.
- **Création de règles :** Vous définissez des règles personnalisées pour bloquer des attaques spécifiques.
- **Logs et débogage :** Les logs d'audit fournissent des détails sur chaque requête suspecte.
- **CRS :** Les règles OWASP offrent une protection robuste pour démarrer rapidement.