Voici une démonstration complète pour protéger un serveur **Apache** contre les attaques en **brute force** en utilisant des outils tels que **Fail2Ban** et **mod_security**, avec des tests pour valider la configuration.

---

1. Comprendre l'attaque par brute force

Une attaque par **brute force** consiste à tenter de deviner des informations sensibles (comme des mots de passe ou des identifiants) en essayant un grand nombre de combinaisons. Les cibles typiques incluent :

- Pages de connexion.
- Directoires protégés par HTTP Basic Auth.
- Services comme WordPress, PHPMyAdmin, etc.

---

2. Mesures pour prévenir les attaques de brute force dans Apache

Pour sécuriser Apache :

1. **Limiter les tentatives de connexion** en bannissant les IP suspectes.
2. **Détecter les requêtes répétées suspectes**.
3. **Ajouter une couche de protection supplémentaire** sur les zones sensibles.

---

3. Étape 1 : Configurer Apache Basic Auth

Si vous utilisez Apache pour protéger une zone par HTTP Basic Auth, vous pouvez configurer un accès restreint.

Configurer l'authentification Basic

1. Créez un fichier de mots de passe pour un utilisateur :

   ```
   sudo htpasswd -c /etc/apache2/.htpasswd user1
   ```

   (Remplacez `user1` par le nom de l'utilisateur souhaité.)

2. Configurez un VirtualHost ou un répertoire avec Basic Auth dans Apache :
   Éditez votre configuration dans `/etc/apache2/sites-available/secure-site.conf` :

   ```
   <Directory "/var/www/secure">
       AuthType Basic
       AuthName "Restricted Access"
       AuthUserFile /etc/apache2/.htpasswd
       Require valid-user
   </Directory>
   ```

3. Redémarrez Apache pour appliquer la configuration :

```
sudo systemctl restart apache2
```

Testez l'accès protégé :

Accédez à `http://example.com/secure`. Vous serez invité à entrer un identifiant et un mot de passe.

---

4. Étape 2 : Détection et blocage avec Fail2Ban

Fail2Ban peut surveiller les logs d'Apache pour détecter les tentatives de brute force.

Configurer Fail2Ban pour Apache

1. Créez ou éditez le fichier `/etc/fail2ban/jail.local` :

```
sudo nano /etc/fail2ban/jail.local
```

2. Ajoutez une configuration pour Apache Basic Auth :

```
[apache-auth]
enabled = true
port = http,https
filter = apache-auth
logpath = /var/log/apache2/error.log
maxretry = 5
findtime = 300
bantime = 3600
```

   - **`maxretry`** : Nombre maximum de tentatives avant de bannir une IP.
   - **`findtime`** : Temps en secondes pendant lequel les tentatives sont comptées.
   - **`bantime`** : Temps pendant lequel l'IP est bannie.

3. Créez un filtre Fail2Ban pour Apache Auth :
   Éditez le fichier `/etc/fail2ban/filter.d/apache-auth.conf` :

```
[Definition]
failregex = ^<HOST> - .* "GET .*" 401
ignoreregex =
```

   - **`401`** : Code HTTP pour les tentatives d'accès non autorisées.

4. Redémarrez Fail2Ban :

```
sudo systemctl restart fail2ban
```

---

5. Étape 3 : Ajouter une Protection avec `mod_security`

`mod_security` peut être configuré pour détecter et bloquer les tentatives de brute force.

Configurer `mod_security` pour détecter les requêtes répétées

1. Ajoutez des règles dans `/etc/modsecurity/custom-rules.conf` :

```
SecAction "id:2001,phase:1,nolog,pass,initcol:ip=%
{REMOTE_ADDR},setvar:ip.login_failures=0"

SecRule RESPONSE_STATUS "@streq 401"
"id:2002,phase:5,pass,setvar:ip.login_failures=+1"

SecRule IP:LOGIN_FAILURES "@ge 5"
"id:2003,phase:1,deny,status:403,msg:'Too many failed login
attempts'"
```

   - `login_failures` : Compteur des échecs de connexion par IP.
   - `@ge 5` : Si le compteur atteint 5, la requête est bloquée.

2. Incluez ce fichier dans la configuration principale de `mod_security` :

```
Include /etc/modsecurity/custom-rules.conf
```

3. Rechargez Apache :

```
sudo systemctl reload apache2
```

---

6. Étape 4 : Tests Pratiques

Test 1 : Simulation de Brute Force avec Curl

Envoyez des requêtes répétées avec des identifiants incorrects :

```
for i in {1..10}; do
    curl -u user1:wrongpassword http://example.com/secure
done
```

- **Résultat attendu avec `mod_security`** :
  Après 5 tentatives, les requêtes sont bloquées avec un statut **403 Forbidden**.
- **Résultat attendu avec Fail2Ban** :
  L'IP est bannie après 5 tentatives échouées.

Test 2 : Vérifier les logs

1. **Logs Apache :**
   Les tentatives échouées apparaissent dans le fichier `/var/log/apache2/error.log` :

   ```
   [client 192.168.1.100] user user1: authentication failure for
   "/secure": Password Mismatch
   ```

2. **Logs Fail2Ban :**
   Vérifiez les IP bannies :

   ```
   sudo fail2ban-client status apache-auth
   ```

3. **Logs `mod_security` :**
   Consultez le fichier `/var/log/apache2/modsec_audit.log` pour voir les requêtes bloquées.

---

7. Étape 5 : Bonnes Pratiques

1. **Limiter l'accès aux zones sensibles par IP** :
   Configurez Apache pour autoriser uniquement des plages d'IP spécifiques :

   ```
   <Directory "/var/www/secure">
       Require ip 192.168.1.0/24
   </Directory>
   ```

2. **Ajouter un CAPTCHA** :
   Intégrez un CAPTCHA sur la page de connexion pour éviter les automatisations.

3. **Combiner avec un WAF externe** :
   Utilisez des solutions comme Cloudflare pour limiter les tentatives de brute force avant qu'elles n'atteignent votre serveur.

---

8. Résumé

1. **Apache Basic Auth** protège les zones sensibles.
2. **Fail2Ban** détecte et bannit les IP effectuant des tentatives répétées.
3. **mod_security** bloque les requêtes malveillantes après plusieurs échecs.
4. **Tests avec curl** confirment que la configuration fonctionne.