Apache2

- Installer un Reverse Proxy Apache2 sur Debian 11
- Sécuriser Apache2 sur un serveur
- Debian 11 utiliser Certbot avec Apache2
- Certificat SSL Apache2 Debian 11
- Installer CrowdSec pour Apache sur Debian 11
- Créer une Autorité de certification

Installer un Reverse Proxy Apache2 sur Debian 11

Dans cette procédure je vais vous expliquer comment installer un serveur **reverse proxy** sur Debian 11 avec **Apache2**. Un serveur Reverse Proxy est un serveur qui se situe entre un accès internet et les serveurs internes afin de gérer une mémoire cache des applications.

Prérequis pour installer un Reverse Proxy Apache2 :

- Une machine sous Debian 11
- Un DNS Local ou public
- Un serveur web

Installer un Reverse Proxy Apache2 sur Debian 11 :

Depuis une machine Debian 11 qui ne va servir qu'a ça (recommandation mais pas obligé), nous allons saisir les instructions suivantes :

Mise à jour du système :

```
apt update && apt full-upgrade -y
```

Installation des **Apache2** :

```
apt-get install apache2
```

Afin de bénéficier du mode reverse sur Apache2, il faut activer les module : proxy et proxy_http. Pour activer ces 2 modules saisissez la commande :

```
a2enmod proxy proxy_http
```

Pour que l'activation de ces modules soient pris en compte redémarrer le service d'apache2 :

```
systemctl restart apache2
```

Créer un fichier de configuration Apache :

```
vim /etc/apache2/conf-available/votre-conf.conf

# ou
nano /etc/apache2/conf-available/votre-conf.conf
```

Voici un fichier de configuration d'exemple :

```
<VirtualHost *:80>
    ServerName votre-domaine.fr
    ServerAdmin postmaster@domaine.fr

    ProxyPass / http://127.0.0.1/
    ProxyPassReverse / http://127.0.0.1/
    ProxyRequests Off
</VirtualHost>
```

- ServerName correspond à votre domaine
- ProxyPass et ProxyPassReverse correspondent au serveur de destination.
- ProxyRequests est en off pour des raison de sécurité.

Activer la configuration :

```
a2ensite votre-conf.conf
```

Puis redémarrer apache2 :

```
systemctl restart apache2
```

> Source :
>
> *https://httpd.apache.org/docs/2.4/fr/*

Recommandations :

Je vous recommande de sécuriser ce serveur à l'aide d'un firewall sur vos deux machines, comme UFW qui est facile à prendre en mains ou Iptables. Sur votre machine qui sert de serveur Web, vous pouvez autoriser uniquement le serveur reverse proxy à se connecter à l'aide du port 80 ou 443 (ports part défaut) et n'oubliez pas de garder le port SSH (port 22 par défaut) ouvert si vous l'utilisez. Vous pouvez également utiliser l'outil CrowdSec afin de limiter les tentatives de cyberattaques si votre site web est exposé sur internet.

Je vous recommande aussi d'utiliser faill2ban pour limiter les attaques de force brute sur vos serveurs web Apache2.

Sécuriser Apache2 sur un serveur

Dans cette procédure, je vais vous montrer quelques bases pour sécuriser vos serveurs Apache2. Je vous recommande de faire une sauvegarde de votre serveur avant toutes modifications, quelques opérations peuvent être sensibles pour serveur Apache2.

Prérequis pour sécuriser Apache2 sur un serveur :

- Un serveur Apache2
- Avoir fait une sauvegarde de ce serveur

Modifications sur la configuration Apache2 :

Pour sécuriser un serveur Apache2, la première étape à faire sur un serveur Apache2. C’est de cacher la version du service.

Not Found

The requested URL /asdf was not found on this server.

Apache/2.4.7 (Ubuntu) Server at 127.0.0.1 Port 80

Not Found

The requested URL was not found on this server.

Le but va être de passer de la première image à la deuxième. Pour cela il faut modifier la configuration de Apache2 :

```
vim /etc/apache2/conf-enabled/security.conf
```

Remplacer : ServerSignature On
Par : ServerSignature Off

Il est également possible de faire cette étape avec la commande suivante :

```
sed -i 's/ServerSignature On/ServerSignature Off/g' /etc/apache2/conf-enabled/security.conf
```

La seconde étape de sécurisation du serveur Apache2 sera de désactiver la lisibilité des fichiers présents dans les dossiers :

```
vim /etc/apache2/conf-enabled/security.conf
```

Ajouter ceci à la fin du fichier :

```
<Directory /var/www>
□□Options -Indexes
</Directory>
```

Index of /assets/css

Name	Last modified	Size	Description
Parent Directory		-	
bootstrap-grid.css	2019-02-13 14:47	63K	
bootstrap-grid.css.map	2019-02-13 14:47	148K	
bootstrap-grid.min.css	2019-02-13 14:47	47K	

Forbidden

You don't have permission to access this resource.

Maintenant, il faut redémarrer le service Apache2 pour que la configuration soit prise en compte :

```
sudo systemctl restart apache2
```

Ensuite avoir un système à jour permet de limiter les possibles intrusions.

```
sudo apt update && sudo apt full-upgrade -y
```

Il est aussi recommandé d'installer un certificat et de forcer la communication avec le protocole HTTPS afin de chiffrer chaque communication.

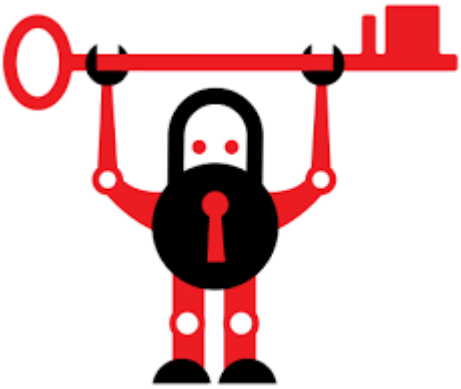
Je recommande l'utilisation de firewall comme Iptables ou UFW. Attention à ne pas bloquer les ports que vous utilisez. Comme le 80/443 pour l'utilisation web et le 22 pour l'utilisation du protocole SSH. Si vous n'avez pas changé les ports par défaut.

Je conseille aussi d'utiliser une machine en tant que reverse proxy dans le but de limiter les possibles accès à vos données qui sont accessibles depuis les serveurs web que vous possédez. De plus, en cas de Cyber attaque il s'agira du serveur qui contiendra ce reverse proxy qui sera le premier à subir l'attaque. Il vous offrira aussi un cache permettant d'obtenir de meilleur temps de réponses lors des chargements des pages de votre site Web. Attention, il ne faut pas abuser de ce cache et le nettoyer régulièrement sinon il pourrait être source d'intrusion.

Il est également recommandé d'utiliser un parseur de logs (comme CrowdSec) afin de détécter les comportements annormaux : **Installer CrowdSec pour Apache sur Debian 11**

Debian 11 utiliser Certbot avec Apache2

Dans cet article, je vais vous montrer comment utiliser Certbot avec Debian 11 pour mettre en place un certificat SSL (Let’s encrypt) pour votre site web. Dans cette procédure, je vais utiliser Apache2 comme serveur web, si vous souhaitez utiliser un autre type de serveur il faudra adapter les commandes. Avoir un certificat SSL sur votre site web vous permet de chiffrer la communication en votre serveur Web et le client, en revanche le certificat SSL sur un site garantie pas que le site est sécurisé.



Prérequis :

- Une machine Debian avec un accès root

Utiliser Certbot sur Debian 11 pour mettre en place un certificat SSL:

Pour utiliser Cerbot sur une machine Debian 11 (avec Apache2), il faut d’abord l’installer :

On va d’abord mettre à jour la liste des paquets :

```
apt update
```

Puis nous allons télécharger Certbot :

```
apt install certbot python3-certbot-apache -y
```

Ensuite vous allez avoir accès à la commande Certbot pour générer les certificats et Certbot s'occupe de mettre en place les certificats :

```
certbot --apache
```

Puis répondez aux questions demandées par Certbot, ensuite redémarrer Apache si vous utilisez Apache ou sinon le service web que vous utilisez :

```
systemctl restart apache2
```

Une fois le service redémarré, le certificat SSL sera installé pour votre site (le virtualhost sera configuré par Certbot).

Si vous souhaitez utiliser Certbot pour un sous-domaine vous pouvez utiliser la commande suivante :

```
certbot --apache --expand -d domaine.com  -d sous.domaine.com
```

> Sources :
>
> *https://certbot.eff.org/*

Certificat SSL Apache2 Debian 11

Le certificat SSL est important pour un serveur Web car il va crypté chaque communication avec le serveur Web et le Client. Pour avoir le HTTPS sur son serveur Web Apache2 qui se trouve sur la distribution Debian 11, il faut installer un certificat SSL. Dans cette procédure, je vais le réaliser à l'aide d'un VPS, un DNS et un certficat SSL de l'hébergeur 1&1 Ionos.

Prérequis pour un certificat SSL dans Apache2:

- Un VPS sous Debian 11
- Un DNS
- Un Certificat SSL (certficat.cert, certificat_intermédiaire.cert, cle_privee.key)
- Avoir installer Apache2

Installer un certificat SSL dans Apache2 :

Pour installer le certificat, utiliser un protocole de transfert de fichiers pour envoyer sur votre serveur les fichiers.

Nous allons créer plusieurs répertoire, chacun devra contenir un fichier.

```
mkdir -p /etc/ssl/cles
mkdir -p /etc/ssl/certificats
```

Dans le dossier « cles » on va y mettre le fichier de clé privées (.key)

Dans le dossier « certificats », il faut mettre les fichiers certificats (.cert), le certificat et le certificat intermédiaire.

Puis rendez-vous dans le fichier de configuration du VirtualHost de votre site à l'emplacement suivant /etc/apache2/sites-available/

Modifier votre VirtualHost :

```
<VirtualHost *:80>
        ServerName votre-domaine.fr

        ServerAdmin postmaster@votre-domaine.fr
        DocumentRoot /var/www/votre-domaine

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf

        Redirect permanent / https://votre-domaine.fr
</VirtualHost>
```

```
<VirtualHost *:443>
        ServerName votre-domaine.fr
        ServerAdmin postmaster@votre-domaine.fr
        DocumentRoot /var/www/votre-domaine

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on
        SSLCertificateFile /etc/ssl/certificats/votre-domaine.fr_ssl_certificate.cer
        SSLCertificateKeyFile /etc/ssl/cles/votre-domaine.fr_private_key.key
        SSLCertificateChainFile /etc/ssl/certificats/votre-domaine.fr_ssl_certificate_INTERMEDIA
</VirtualHost>
```

Une fois la configuration du VirtualHost faite et enregistrer, vous pouvez vérifier que votre configuration apache2 est correct :

Si vous n'aviez pas de VirtualHost avant, n'oubliez pas de l'activée ! Avec la commande a2ensite votre-domaine.fr

```
apachectl configtest
```

Si le message suivant vous est retourner, c'est que votre configuration est correct sinon il faut regarder votre fichier de configuration, il doit y avoir une erreur.

```
root@localhost:/etc/apache2/sites-available# apachectl configtest
Syntax OK
```

Une fois que vous êtes sûr que votre configuration est correct, alors vous pouvez redémarrer le service Apache2.

```
systemctl restart apache2
```

Après avoir fait cette opération si vous essayé de vous connecter sur votre domaine depuis un navigateur. Vous aller être automatiquement rediriger sur le protocole HTTPS, même si vous précisé que vous utilisé le protocole HTTP.

> Sources :
>
> *https://wiki.debian.org/Apache*

Installer CrowdSec pour Apache sur Debian 11

Dans cette procédure, je vais vous montrer comment installer CrowdSec sur Debian 11 pour protéger son site Web Apache. CrowdSec est un outil qui va permettre de bloquer les attaques qu'il détecte sur le serveur, nous allons utilisé un serveur Debian 11 dans notre démonstration. CrowdSec fonctionne avec des modules appelés des Bouncers. Les bouncers permettent de bloqué les trafics réseau qui ne sont pas détecter comme « normaux ». Ils peuvent soit bloqué complètement la requête ou afficher à Captcha dans le but de vérifier qu'il ne s'agit pas d'un robot.

Prérequis :

- Une machine Debian avec Apache2 d'installé.
- Avoir Composer d'installer
- Avoir les permissions root ou avoir sudo

Installer CrowdSec pour Apache 2 sur Debian 11 :

Avant de commencer l'installation de CrowdSec, commencer par mettre à jour la liste des paquets votre machine Debian 11

```
sudo apt update
```

Ensuite télécharger le CrowdSec sur votre machine Debian :

```
sudo apt install -y crowdsec
```

Après avoir installer CrowdSec sur votre machine, la CLI de CrowdSec (CSCLI) vous permet de voir les services pouvant être protéger par CrowdSec présents sur votre machine.

```
cscli collections list
```

```
root@localhost:~# cscli collections list
--------------------------------------------------------------------------------------------------------------
 NAME                                  STATUS    VERSION  LOCAL PATH
--------------------------------------------------------------------------------------------------------------
 crowdsecurity/modsecurity             ✔ enabled  0.1      /etc/crowdsec/collections/modsecurity.yaml
 crowdsecurity/vsftpd                  ✔ enabled  0.1      /etc/crowdsec/collections/vsftpd.yaml
 crowdsecurity/sshd                    ✔ enabled  0.1      /etc/crowdsec/collections/sshd.yaml
 crowdsecurity/base-http-scenarios     ✔ enabled  0.4      /etc/crowdsec/collections/base-http-scenarios.yaml
 crowdsecurity/mysql                   ✔ enabled  0.1      /etc/crowdsec/collections/mysql.yaml
 crowdsecurity/linux                   ✔ enabled  0.2      /etc/crowdsec/collections/linux.yaml
 crowdsecurity/postfix                 ✔ enabled  0.2      /etc/crowdsec/collections/postfix.yaml
 crowdsecurity/whitelist-good-actors   ✔ enabled  0.1      /etc/crowdsec/collections/whitelist-good-actors.yaml
 crowdsecurity/apache2                 ✔ enabled  0.1      /etc/crowdsec/collections/apache2.yaml
 crowdsecurity/dovecot                 ✔ enabled  0.1      /etc/crowdsec/collections/dovecot.yaml
 crowdsecurity/nginx                   ✔ enabled  0.1      /etc/crowdsec/collections/nginx.yaml
 crowdsecurity/wordpress               ✔ enabled  0.1      /etc/crowdsec/collections/wordpress.yaml
 crowdsecurity/iptables                ✔ enabled  0.1      /etc/crowdsec/collections/iptables.yaml
 crowdsecurity/naxsi                   ✔ enabled  0.1      /etc/crowdsec/collections/naxsi.yaml
--------------------------------------------------------------------------------------------------------------
```

Vous pouvez également lister les bouncers installés et disponibles sur votre machine :

```
cscli bouncers list
```

Si vous exécutez cette commande, vous allez voir un tableau vide. C’est normal car nous n’avons pas encore installer de bouncer.

Installation du Bouncer Apache 2 sur CrowdSec (Debian 11) :

Avant de commencer l’installation du Bouncer, assurez-vous d’avoir composer d’installé. Si vous ne l’avez pas installer, voici une procédure pour l’installer :

Installer Composer sur Debian 11

Nous allons installer git sur la machine pour pouvoir cloner le Repo du bouncer :

```
sudo apt install git -y
```

Puis nous clonons le Repo GitHub :

```
git clone https://github.com/crowdsecurity/cs-php-bouncer.git
```

On se rend dans le dossier du projet :

```
cd cs-php-bouncer/
```

Enfin on lance l'installation avec un compte utilisateur (pas avec le compte root ni avec sudo mais le compte utilisateur doit être dans le groupe 'sudo')

```
./install.sh --apache
```

Puis on donne les droit en tant que propriétaire www-data (apache) sur le répertoire de CrowdSec

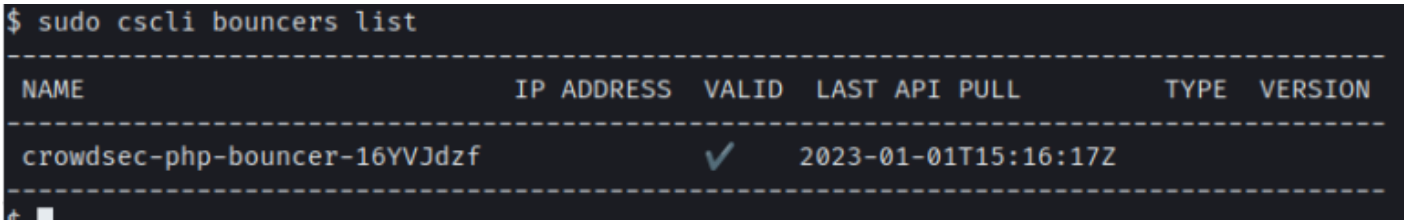
```
sudo chown www-data /usr/local/php/crowdsec/
```

Puis on redémarre le service de Apache

```
sudo systemctl reload apache2
```

Enfin le Bouncer de Apache va apparaître dans la liste des bouncers :

```
sudo cscli bouncers list
```



Ensuite on va afficher un Captcha pour les utilisateurs qui consulterons notre site web avec un mauvais User-agent (Généralement les outils de pentest utilisent des User-Agent différent des navigateurs Web) et les crawler non static.

On va modifier le fichier de configuration :

```
sudo nano /etc/crowdsec/profiles.yaml
```

```
# Mauvais User-Agent + Crawler
name: crawler_captcha_remediation
filters:
 - Alert.Remediation == true && Alert.GetScenario() in ["crowdsecurity/http-crawl-non_statics",
decisions:
 - type: captcha
   duration: 4h
on_success: break
---
name: default_ip_remediation
filters:
 - Alert.Remediation == true && Alert.GetScope() == "Ip"
decisions:
```

```
 - type: ban
   duration: 4h
on_success: break
---
```

Puis on redémarre CrowdSec pour qu'il utilise la nouvelle configuration :

```
sudo systemctl restart crowdsec
```

Il est possible d'afficher toutes les actions que CrowdSec aura pris avec cette commande :

```
sudo cscli decisions list
```

C'est également possible de débanir une adresse IP :

```
sudo cscli decisions delete --ip adresse-ip-a-deban
```

La CLI de CrowdSec est pratique mais CrowSec propose également une interface graphique Web avec Docker.

> Sources :
>
> *https://doc.crowdsec.net/*

Créer une Autorité de certification

Dans cette procédure je vais vous expliquer comment créer une autorité de certification et un certificat hôte pour Apache2. Pour réaliser l'autorité de certification nous allons utiliser Openssl et une machine Debian.

Prérequis :

- Une machine sous Debian
- Openssl

Créer une Autorité de certification :

D'abord, nous allons créer un répertoire pour notre autorité de certification :

```
mkdir -p /etc/ssl/certificats/CA
```

Ensuite nous allons créer la clé privée de l'autorité de certification :

```
openssl genrsa -des3 -out /etc/ssl/certificats/CA/CA.key 2048
```

Saisir une passphrase qui sera utilisée pour signer les certificats (Donc à ne pas perdre).

Puis nous allons générer le certificat *root* (racine) de l'autorité de certification au format .pem :

```
 openssl req -x509 -new -nodes -key /etc/ssl/certificats/CA/CA.key -sha256 -days 10000 -out /etc/ssl/certificats/CA/CA.pem
```

Une liste de questions va vous être demandée.

Ensuite nous allons générer le certificat root (racine) au format .crt :

```
openssl x509 -in /etc/ssl/certificats/CA/CA.pem -inform PEM -out /etc/ssl/certificats/CA/CA.crt
```

Installer l’autorité de certification sur une machine :

Nous avons créer 3 fichiers dans le répertoire /etc/ssl/certificats

Intallation version web :

Le fichier CA.crt est a importé dans votre navigateur Web :

- Firefox => saisir dans l’url : about:preferences#privacy => Se rendre dans **Certificats** => **Afficher les certificats** => **Importer** => Choisir le fichier .crt
- Chrome => saisir dans l’url : chrome://settings/security => Se rendre dans **Gérer les certificats** => **Importer** => Choisir le fichier .crt

Installation au niveau de l’OS :

Windows :

Récupérer le fichier CA.crt et double cliquer dessus => Choisir l’emplacement : « Autorité de certification racine de confiance »

Ou en ligne de commande :

certutil.exe -addstore root CA.crt

Pour Firefox, il faut autoriser l’utilisation des autorités de certifications de confiances de Windows. Donc il faut créer un fichier dans le répertoire suivant C:\Program Files (x86)\Mozilla Firefox\Defaults\Pref\defaults\pref\ ou C:\Program Files\Mozilla Firefox\Defaults\Pref\defaults\pref\ :

Créer un fichier enableroot.js
Ajoutez y le contenu suivant :

```
pref("security.enterprise_roots.enabled", true);
```

Linux :

D’abord, on va copier le fichier CA.crt dans le répertoire des autorités (**/usr/local/share/ca-certificates/**) :

```
cp CA.crt /usr/local/share/ca-certificates/
```

Ensuite mettre à jour les autorités :

```
update-ca-certificates
```

Créer un certificat hôte :

D’abord nous allons créer un répertoire pour notre hôte :

```
mkdir -p /etc/ssl/certificats/hote
```

Ensuite nous allons créer la clé privée de l’hôte :

```
openssl genrsa -out /etc/ssl/certificats/hote/hote.key 2048
```

Puis on génère la demande de signature de certificat (fichier au format .csr) :

```
openssl req -new -key /etc/ssl/certificats/hote/hote.key -out
/etc/ssl/certificats/hote/hote.csr
```

Dans les questions qui sont demandées, il faut mettre le DNS du serveur pour le Common Name.

Signer le certificat hôte par l’autorité de certification :

D’abord on va créer un fichier de configuration :

```
nano /etc/ssl/certificats/hote/hote.ext
```

Avec le contenu suivant :

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names
[alt_names]
DNS.1 = hote
```

Pour signer le certificat de l'hôte par l'autorité de certification précédemment créée nous allons exécuter la commande suivante :

```
openssl x509 -req -in /etc/ssl/certificats/hote/hote.csr -CA /etc/ssl/certificats/CA/CA.pem -CAk
```

Enfin on renseigne la passphrase de l'autorité de certification.

Intégré le certificat signé avec Apache2 :

Pour intégré le certificat dans Apache2, on doit éditer le Virtual Host qui est utilisé dans le répertoire **/etc/apache2/sites-available** :

```
nano /etc/apache2/sites-available/000-default.conf
```

Voici un exemple de configuration :

```
<VirtualHost *:80>
        ServerName hote

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        Redirect permanent / https://hote/
</VirtualHost>
<VirtualHost *:443>
        ServerName hote

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
        SSLEngine on
        SSLCertificateFile /etc/ssl/certificats/hote/hote.crt
        SSLCertificateKeyFile /etc/ssl/certificats/hote/hote.key
</VirtualHost>
```

Cette configuration permet d’automatiquement rediriger les connexions http en https et d’utiliser les fichiers du certificat et de la clé de l’hôte.

Ensuite on va activer le SSL sur Apache2 :

```
a2enmod ssl
```

Puis on redémarrer le service Apache2 :

```
systemctl restart apache2
```