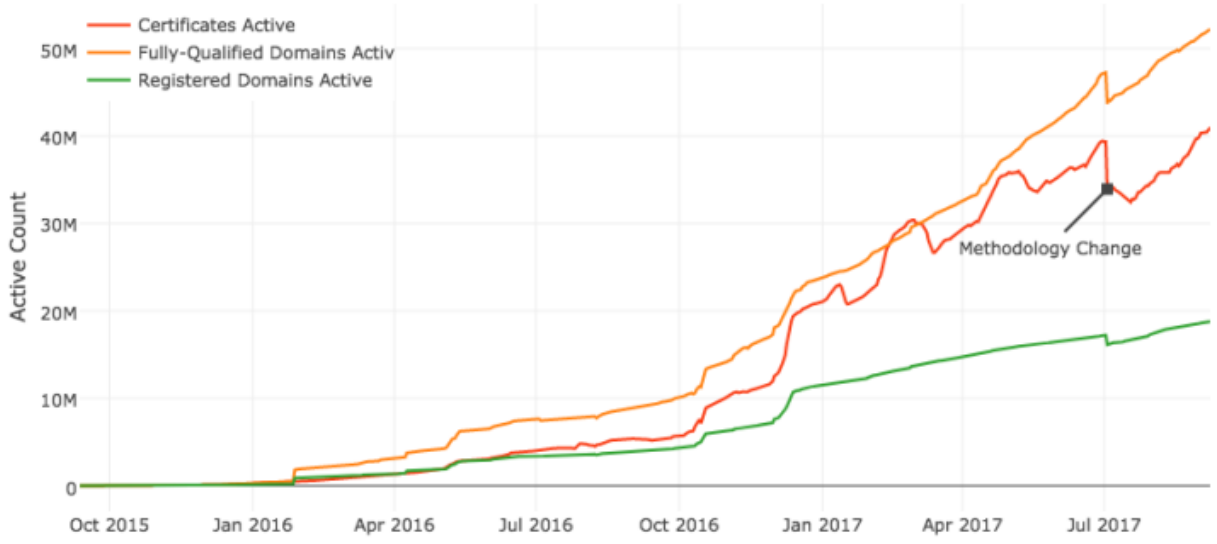
# Sécuriser les connexions d'apache avec let's encrypt

Let's Encrypt est une autorité de certification lancée le 3 décembre 2015 (Bêta Version Publique) par l'internet Security Research Group.
Cette autorité fournit des certificats gratuits X.509 pour le protocole cryptographique TLS.
Le projet généralise l'usage des connexions sécurisées sur internet en supprimant la nécessité de paiement, de configuration du serveur web, des mails de validation et de renouvellement des certificats.
Il réduit donc, la complexité de mise en place et de maintenance du chiffrement TLS.
Pour cela, il faut installer un logiciel (certbot) qui va permet la gestion de certificats pour un serveur web, il est écrit en Python et tous cela au travers du protocole ACME, qui permet l'automatisation des échanges entre l'autorité de certification et le propriétaire du serveur web.
Aujourd'hui, il y a plus de 40 millions de certificats actif et plus de 18 millions de domaine enregistré chez let's encrypt.




Donc vous l'aurez compris, je vais donc mettre en place un certificat TLS avec let's encrypt.

## spécifications :

Vous devrez disposer des choses suivantes :

- Un nom de domaine public enregistré avec des enregistrements A valides pour indiquer l'adresse IP externe de votre serveur.
- Dans le cas où votre serveur est derrière un pare-feu, prenez les mesures nécessaires pour vous assurer que votre serveur est accessible sur internet en ajoutant des règles de transfert de port du côté du routeur.
- Le serveur web Apache est installé avec le module SSL activé et l'hôte virtuel est activé, dans le cas où vous hébergez plusieurs domaines ou sous-domaines.

## Installation d'apache et activation du module SSL

Si apache n'est pas installé sur votre machine, voici la commande sur Debian ou équivalent :

```
apt-get install apache2
```

Il faudra activer le module ssl de apache, puis nous allons activer notre hôte virtuel :

```
a2enmod ssl
a2ensite default-ssl.conf
```

Puis, nous redémarrons le service apache :

```
service apache2 restart
```

## Installation du certbot

Vous aurez besoin de `git` pour installer le cerbot :

```
apt-get install git
```

J'ai choisis de cloner le certbot dans `/usr/local`, vous pouvez choisir de le faire ailleurs :

```
cd /usr/local
git clone https://github.com/letsencrypt/letsencrypt
```

## Génération d'un certificat pour apache

Dans de nombreux cas, vous pouvez simplement exécuter `certbot-auto` ou `certbot` et le client vous guidera dans le processus d'obtention et d'installation de certificats de manière interactive.

Pour l'aide en ligne de commande complète, vous pouvez taper:

```
./certbot-auto --help all
```

Vous pouvez également lui dire exactement ce que vous voulez faire à partir de la ligne de commande.

Par exemple, si vous souhaitez obtenir un certificat pour `domaine.com`, `autres.domaine.com` et `xxx.domaine.org`, en utilisant le plugin Apache pour obtenir et installer les certificats, vous pouvez le faire :

```
./certbot-auto --apache -d domaine.com -d autres.domaine.com xxx.domaine.org
```

Il vous faudra suivre les instructions, et vous posséderez vos certificats.

Vos certificats sont stockés dans `/etc/letsencrypt/live`.

## renouvellement

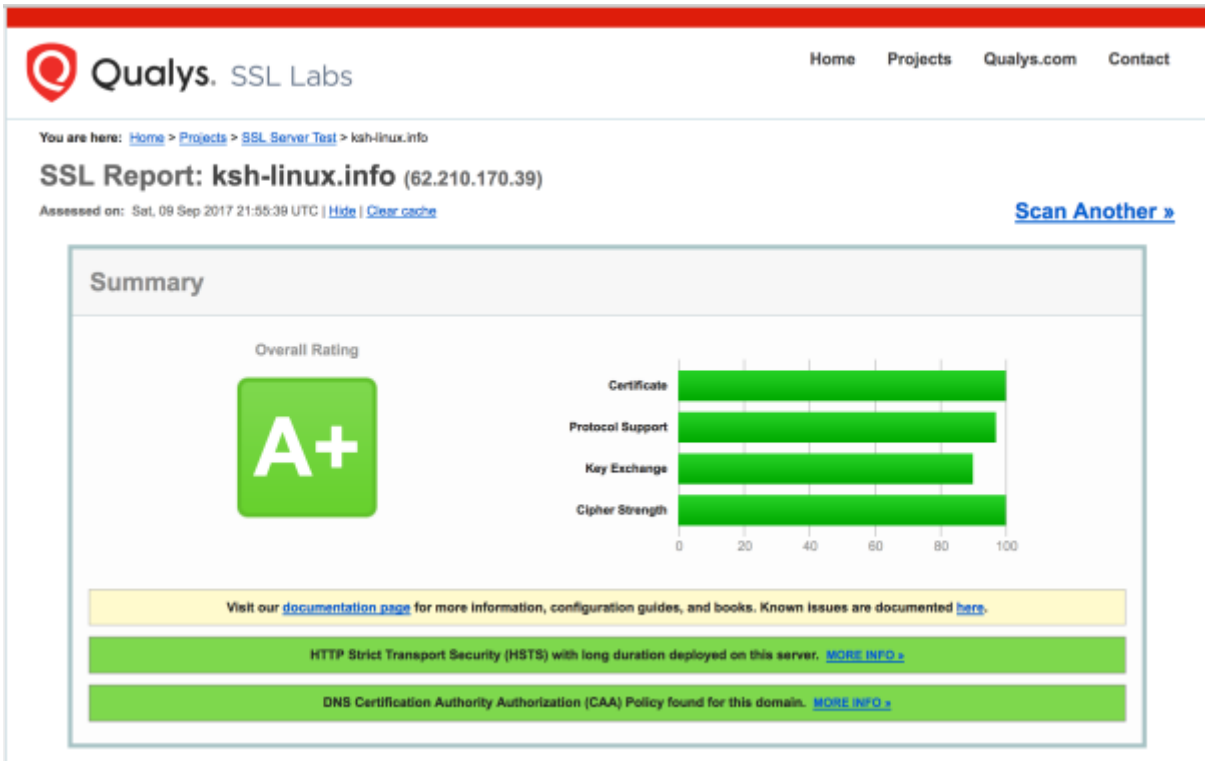
L'une des choses importante à savoir avec let's encrypt est qu'un certificat généré a une durée de validité de 90 jours ce qui est court, heureusement sous linux nous pouvons gérer cela avec une crontab :

```
crontab -e
```

Voici, ce que vous ajouterez :

```
0 1 1 */2 /usr/local/letsencrypt/certbot-auto -n renew
```

Vous pouvez test votre site sur ssllabs.com, voici mon test:




## Comment améliorer cela

Ajouter dans votre domaine public une enregistrement CAA, cela va permettre d'indiquer aux autorités de certification quelles sont celles autorisées à certifier le domaine.
Dans sa forme la plus simple, l'enregistrement ressemble en général à l'exemple ci-dessous.
L'autorité let's encrypt (et seulement elle) est autorisée à émettre des certificats pour le nom de domaine example.com, ainsi que pour www.example.com et www.subdomain.example.com grâce au mécanisme hiérarchique de vérification :

```
exemple.com     CAA         0    issue    "letsencrypt.org"
```

Les enregistrements de ressource (RR) CAA sont pris en charge par le serveur DNS BIND versions 9.10.1B et ultérieures

Dans apache, nous allons mettre en place certaines options pour :

- faire en sorte que la préférence du serveur soit choisis lors du chiffrement
- préciser une chaîne spécifique de chiffrement
- Spécifier les protocoles que nous souhaitons utiliser
- Désactiver la compression SSL pour éviter certains problème

Voici, donc les options à ajouter dans votre hôte virtuel :

```
SSLHonorCipherOrder On
SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-
SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-
ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-
SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-
SHA256
SSLProtocol  ALL -TLSv1 -TLSv1.1 -SSLv2 -SSLv3
SSLCompression Off
```

Voilà, votre site possède une certificat chez Let's encrypt, Cool, qu'est-ce que ça vous fait.

source: wikipédia.org