Let's Encrypt

Comprendre Let's Encrypt

Let's Encrypt est un Autorité de Certification qui est **GRATUITE** et qui permet de signer un certificat de façon automatique et quasi instantané.

Deux méthodes existent pour valider un certificat :

1. Par requête HTTP : Lors de la demande de certificat, l'API va effectuer une requête HTTP sur la dites URL dans le répertoire */.well-known/acme-challenge* d'un fichier très spécifique qu'il aura demandé de créer lors de la dites requête.
2. Par requêtes DNS : Lors de la requête, l'API fera une requête TXT particulière sur le DNS. Celle-ci n'est soit pas automatique ou elle oblige que le serveur WEB puissent mettre à jour sa zone DNS. Bien que plus contraignante, elle permet depuis la version 2 de signer des certificats WildCard.

Pourquoi cette méthode

L'objectif de cette page est de générer et faire signer des certificats avec Let's Encrypt.
Cette méthode n'est pas la plus simple, mais elle permet de mettre en place DANE/TLSA et HPKP ainsi que d'avoir des certificats en ECC.

Pour cela, il faut juste le packet certbot/letsencrypt.

Cette méthode n'est pas éligible au renouvellement automatique de certbot, donc il faudra ajouter manuellement l'entrée en tache cron. La méthodologie, si suivie a la lettre, permet un gestion simple de ces certificats.

Génération du certificat

Pour chaque domaines, nous allons créer les dossiers où seront stocké les certificats et les clés:

```
# mkdir -p /etc/ssl/custom-certbot/{live,archive}/wiki.virtit.fr
```

puis on génère la clé privé:

```
# openssl ecparam -name prime256v1 -genkey -out /etc/ssl/custom-
certbot/live/wiki.virtit.fr/privkey.pem
```

Puis le certificat publique

```
# openssl req -new -subj "/CN=wiki.virtit.fr" -key /etc/ssl/custom-
certbot/live/wiki.virtit.fr/privkey.pem -nodes -out /etc/ssl/custom-
certbot/live/wiki.virtit.fr/csr.pem
```

Ces certificats seront unique mais pensez à les renouveler de temps en temps .

Signature du certificat

Pour signer votre certificat, il faut lancer la commande suivante :

```
# certbot certonly --webroot -w /var/www/letsencrypt/ -d wiki.virtit.fr --
csr /etc/ssl/custom-certbot/live/wiki.virtit.fr/csr.pem --cert-path
/etc/ssl/custom-certbot/live/wiki.virtit.fr/cert.pem --chain-path
/etc/ssl/custom-certbot/wiki.virtit.fr/chain.pem --fullchain-path
/etc/ssl/custom-certbot/live/wiki.virtit.fr/fullchain.pem
```

Renouvellement du certificat

Pour renouveler le certificat, j'ai créé un script qui va vérifier la date d'expiration, et renouvelle au besoin.
Je vous conseil d'ajouter la tâche **cron** suivante :

```
0 2 * * * root /opt/renew-cert.sh
```

et créer le script suivant :

/opt/renew-cert.sh

```
#!/usr/bin/env bash

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
cd /etc/ssl/custom-certbot/live/
TEMP=`mktemp -d`
NOW_SECONDS=`date '+%s'`
NEEDTORELOAD=0

for i in * ; do
    END_DATE=`openssl x509 -dates -noout -in ${i}/cert.pem 2>/dev/null
| sed -n 's/ *notAfter=*//p'`
    END_DATE_SECONDS=`date '+%s' --date "$END_DATE"`
    REMAINING_DAYS=`echo "($END_DATE_SECONDS-$NOW_SECONDS)/24/3600" |
bc`
    if [ "$REMAINING_DAYS" -lt "30" ]; then
        NEEDTORELOAD=1
        echo "Renewing $i"
        mkdir ${TEMP}/${i}
        /usr/bin/certbot certonly --webroot -w /var/www/letsencrypt/  -
d $i --csr ${i}/csr.pem --cert-path ${TEMP}/${i}/cert.pem --chain-path
${TEMP}/${i}/chain.pem --fullchain-path ${TEMP}/${i}/fullchain.pem
        EXITCODE=$?
        if [ "$EXITCODE" -eq "0" ]; then
```

```
            if [ ! -d "/etc/ssl/custom-certbot/archive/${i}" ]; then
                mkdir /etc/ssl/custom-certbot/archive/${i}
            fi
            mv ${i}/cert.pem  /etc/ssl/custom-
certbot/archive/${i}/cert.pem
            mv ${i}/chain.pem  /etc/ssl/custom-
certbot/archive/${i}/chain.pem
            mv ${i}/fullchain.pem  /etc/ssl/custom-
certbot/archive/${i}/fullchain.pem
            mv ${TEMP}/${i}/* ${i}/
        fi
    else
        if [ -t 1 ] ; then
            echo "Nothing to do on $i ($REMAINING_DAYS days left)"
        fi
    fi

done

if [ "$NEEDTORELOAD" -eq "1" ]; then
    echo "Reloading services"
    systemctl reload nginx
fi
rm -r $TEMP
```

Il faudra le rendre exécutable, et ajouter les commandes nécessaires au rechargement des services dans le script.