Let's Encrypt Key renewal

> This page is a work in progress!!

This page contains some details on how to renew certificates with Let's Encrypt using the same key, which is very helpful in supporting authentication via a SPKI pinset.

Manual renewal

This assumes that you will use certbot in conjunction with Let's Encrypt and you have an existing key <my_key_file> that was use to sign the previous certificate.

1. Create your new CSR from your existing private key using 'openssl req'

   ```
   openssl req -key <my_key_file> -new -out <my_csr_file>
   ```

2. Use the certbot interface to renew the cert using the same key, for example using web authenticaiton

   ```
   certbot certonly -d <my_authentication_name> --csr <my_csr_file> --
   webroot -w /home/website/public
   ```

   or using dns challenge

   ```
   certbot certonly -d <my_authentication_name> --csr <my_csr_file> --
   preferred_challenges dns --manual
   ```

3. For the **dns challenge mode**, step 2 outputs a TXT file that must be added to the corresponding zone <my_authentication_name> before the certificate can be issued and instructs something like:

   ```
   Please deploy a DNS TXT record under the name
   _acme-challenge.<my_authentication_domain_name> with the following
   value:

   <TXT value>
   Once this is deployed,
   Press ENTER to continue
   ```

   1. Manually add the TXT record and wait until it has propagated e.g. use dig to 8.8.8.8 to obtain the new TXT record.
   2. hit ENTER, which should result in a new certificate being issued.
   3. Restart the nameserver or proxy to have it use the new certification.

Automated renewal

There are a number of ways to do this but one common one is to use https://dehydrated.de/ It is nice for automating the renewal workflow, particularly if you want to use the DNS challenge method, rather than web access. Thanks to Willem Toorop and Ralph Dolmans at NLnet Labs for developing this automated solution!

- An example configuration file is:

```
CA="https://acme-v01.api.letsencrypt.org/directory"
#CA="https://acme-staging.api.letsencrypt.org/directory"
LICENSE="https://letsencrypt.org/documents/LE-SA-v1.1.1-August-1-
2016.pdf"
CERTDIR=/usr/local/etc/dehydrated/certs
CHALLENGETYPE="dns-01"
HOOK=/usr/local/etc/dehydrated/dnshook.sh
PRIVATE_KEY_RENEW="no"
PRIVATE_KEY_ROLLOVER="no"
CONTACT_EMAIL=alice@example.com
```

- Private keys are then stored in

```
/usr/local/etc/dehydrated/certs/<domain>/privkey.pem
```

- The SubjectAltNames are then enumerated in the file

```
/usr/local/etc/dehydrated/domains.txt
```

Add one line in this for each 'group' of names that should share a certificate e.g

```
example.com www.example.com example.org
example1.com www.example1.com example1.net
```

- Then the challenge record needs to be provisioned in the corresponding zone in a record of the form

```
_acme-challenge.<domain name>
```

- If you have many zones it can be helpful to use CNAMES to redirect to a single zone that can hold the acme_challenge records e.g. <domain>.acme.example.com

> The domain *acme.example.com* is then hosted **only** on the server that also runs dehydrated.

- A script can then be used to deploy and clean the challenge in this domain. An example script is included below

```
#!/bin/sh

zonefile=/usr/local/etc/dehydrated/acmezone

deploy_challenge() {
   local DOMAIN="${1}" RDATA="${3}"
   echo "$DOMAIN 10 TXT \"$RDATA\"" >> $zonefile
   echo "$DOMAIN 10 TXT \"$RDATA\""
   ldns-read-zone $zonefile > /dev/null
   if [ $? -eq 0 ]; then
       cp $zonefile ~/unsigned/acme.example.com
       ods-signer sign acme.example.com
       sleep .5
   fi
}

clean_challenge() {
   local DOMAIN="${1}" RDATA="${3}"
   sed -i ".old" "/$DOMAIN 10 TXT \"$RDATA\"/d" $zonefile
   ldns-read-zone $zonefile > /dev/null
   if [ $? -eq 0 ]; then
       cp $zonefile ~/unsigned/acme.example.com
       ods-signer sign acme.example.com
   fi
}

deploy_cert() {
   local DOMAIN="${1}" KEYFILE="${2}" CERTFILE="${3}"
FULLCHAINFILE="${4}" CHAINFILE="${5}" TIMESTAMP="${6}"
   # nothing yet..
}

unchanged_cert() {
   local DOMAIN="${1}" KEYFILE="${2}" CERTFILE="${3}"
FULLCHAINFILE="${4}" CHAINFILE="${5}"
   # nothing yet..
}

HANDLER="$1"; shift
"$HANDLER" "$@"
```