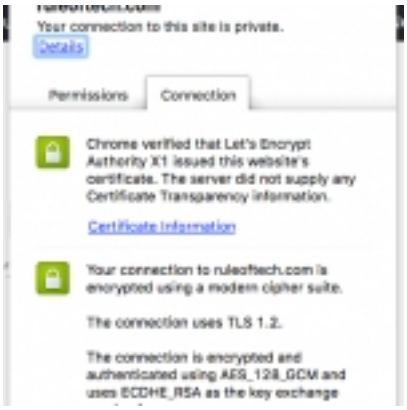

Let's Encrypt is now in public beta, meaning, you can get valid, trusted SSL certificates for your domains for free. Free SSL certificates for everyone! As Let's Encrypt is relatively easy to setup, there's now no reason not to use HTTPS for your sites. The needed steps are described in the documentation and here's short guide how to setup Let's Encrypt in CentOS 6.x and automate the SSL certificate renewal.

Let's Encrypt installation

The Let's Encrypt Client is a fully-featured, extensible client for the Let's Encrypt CA that can automate the tasks of obtaining certificates and configuring web servers to use them. The installation is simple but in my case on CentOS 6.x I first needed to update to Python 2.7 as Let's Encrypt supports Python 2.7+ only.

Installing Python 2.7 in Centos 6.x

Install Epel Repository yum install epel-release # Install IUS Repository rpm -ivh https://rhe

Setting up Lets encrypt

Install Git if you don't have it yet.

```
yum install git
```

If letsencrypt is packaged for your operating system, you can install it from there, and the other solution is to use the letsencrypt-auto wrapper script, which obtains some dependencies from your operating system and puts others in a python virtual environment:

```
# Get letsencrypt git clone https://github.com/letsencrypt/letsencrypt # See help ./letsencrypt
```

Running the client

You can either just run letsencrypt-auto or letsencrypt, and the client will guide you through the process of obtaining and installing certs interactively or you you can tell it exactly what you want it to do from the command line.

For example obtain a cert for your domain using the Apache plugin to both obtain and install the certs, you could do this:

```
./letsencrypt-auto --apache -d thing.com -d www.thing.com -d otherthing.net
```

(The first time you run the command, it will make an account, and ask for an email and agreement to the Let's Encrypt Subscriber Agreement; you can automate those with –email and –agree-tos)

Although you can use the Apache plugin to obtain and install the certs it didn't work for me. I got an error: *"The apache plugin is not working; there may be problems with your existing configuration."* This seems to be an issue with Apache 2.2 and until it's fixed you can use the webroot authentication method as explained in documentation
.

```
./letsencrypt-auto certonly --webroot -w /var/www/example/ -d example.com
```

The webroot plugin works by creating a temporary file for each of your requested domains in *${ webroot-path}/.well-known/acme-challenge*
. Then the Let's Encrypt validation server makes HTTP requests to validate that the DNS for each requested domain resolves to the server running letsencrypt. Note that to use the webroot plugin, your server must be configured to serve files from hidden directories.

Now your certificate and chain have been saved at Let's Encrypt configuration directory at *"/etc/letsencrypt"*
and
"/etc/letsencrypt/live/"
contains symlinks to the latest certificates. Making regular backups of this folder is ideal.

All we have to do now is set it up in Apache.

Configure Apache to use Let's Encrypt certs

In Let's Encrypt configuration directory at *"/etc/letsencrypt/live/"* the .pem files are as follows (from the Letsencrypt documentation):

- privkey.pem: Private key for the certificate.
- This must be kept secret at all times! Never share it with anyone, including Let's Encrypt developers. You cannot put it into a safe, however – your server still needs to access this file in order for SSL/TLS to work.
- This is what Apache needs for SSLCertificateKeyFile

- cert.pem: Server certificate only.
- This is what Apache needs for SSLCertificateFile.

- chain.pem: All certificates that need to be served by the browser excluding server certificate, i.e. root and intermediate certificates only.
- This is what Apache needs for SSLCertificateChainFile.

- fullchain.pem: All certificates, including server certificate. This is concatenation of chain.pem and cert.pem.

Now that we know which file is which we can configure our VirtualHost to use SSL with our new certs. Change the following lines in your Apache's virtualhost's SSL configuration:

... SSLCertificateFile /etc/letsencrypt/live/<your-domain>/cert.pem SSLCertificateKeyFile /etc/le

Finally, restart apache

You can test that your SSL is working with SSL Labs .

Automate updating Let's Encrypt certs

As you surely noticed Let's Encrypt CA issues short lived certificates (90 days) and you have to renew the certificates at least once in 3 months. Nice way to force sysadmins to automate the process.

To obtain a new version of the certificate you can simply run Let's Encrypt again but doing that manually is not feasible. Let's Encrypt is working hard on automating the renewal process but until that we have to do it by ourselves.

Fortunately we don't need to invent our own scripts as there's excellent article about automating Let's Encrypt and script for crontab
.

Get the autole.sh -script from GitHub that automates tasks like:

- Check the expire date of the certificate and renew when the remaining days are below a value
- Check that the directory for the challenge is well mapped
- Alert the admin if it's not possible to renew the certificate

Now you can renew certain domain's certificates with

```
./autole.sh www.mydomain.com
```

And to renew all your certificates use

```
./autole.sh --renew-all
```

Now you can add this to the crontab, run weekly, and your certificates will be ready and renew automatically. This cron job will execute the command every Monday at 08:30.

```
30 8 * * 1 /usr/local/sbin/autole.sh <your-domain> >> /var/log/autole.log
```

Now before I switch my WordPress over to HTTPS I have to do some find & replace in the database and fix the URL’s of the images to be protocol relative.