# 1.3. Manage the Federation Metadata

## 1.3. Manage the Federation Metadata



The Federeation metadata contains a description of how long it may be used, by the attributes cacheDuration and validUntil of the element EntitiesDescriptor (which is normally the first element in the SAML metadata). The Member should normally update his local copy of the Federation Metadata at least with the periodicity that is stated in cacheDuration. SimpleSAMLphp provides the Metararefresh module that automatically updates the metadata. Metadata must not be considered valid after validUntil.

### Metarefresh

#### Eable the metarefresh and cron module

cd /var/simplesamlphp
sudo touch modules/cron/enable
sudo cp modules/cron/config-templates/*.php config/
sudo touch modules/metarefresh/enable
sudo cp modules/metarefresh/config-templates/*.php config/

#### Storing the metadata

Create the directory where the metadata will be stored.
sudo mkdir metadata/federation
sudo chown www-data metadata/federation

#### Verification of the federation operator s public key

To be able to verify the signature in the metadata the federation public key must be used. Get the certificate containing the public key from the federation website. Store the certificate in the directory cert.
When updating the federation operator s public key in a Member s local configuration, the Member must verify its authenticity against at least two different sources. The following are acceptable verification sources:

- fetch the certificate including the public key directly from federation website, including a positive verification of the HTTPS certificate that identifies the site of publication (according to Web PKI)
- contact with the support service, where the certificate s digital SHA-1 fingerprint is verified over telephone

Use openSSL to extract the SHA-1 fingerprint
openssl x509 -noout -in cert/federation.crt -fingerprint -sha1

#### Configure th metarefresh module

Edit config/config-metarefresh.php.

- Set 'src' to the URL of the federations metadata.
- Set 'certificates' to to filename of the federation certificate. The path are configured with the directive 'certdir' in config.php, default cert/
- Set 'exipreAfter' so that i match the metadata attribute validUntil
- Set 'types' to the type of metadata needed.

```
<?php
$config = array(
'sets' => array(
'skolfederation' => array(
'cron' => array('frequent'),
'sources' => array(
array('src' => 'https://federation.se/metadata.xml',
'certificates' => array('federation.crt',),
),
),
'expireAfter' => 60*60*24*3, // Maximum 3 days cache time.
'outputDir' => 'metadata/federation/',
'outputFormat' => 'flatfile',
'types' => array('saml20-idp-remote','saml20-sp-remote'),
),
),
);
```

### Configure SimpleSAMLphp

Edit config/config.php change metadata.source to the following.

```
'metadata.sources' => array(
array('type' => 'flatfile'),
array('type' => 'flatfile', 'directory' => 'metadata/federation'),
),
```

## Configure the cron module

### RANDOM_KEY

Edit the file config/module_cron.php. Change the directive key to a secret value.

```
$config = array (
 'key' => 'RANDOM_KEY',
 'allowed_tags' => array('frequent'),
 'debug_message' => TRUE,
 'sendemail' => TRUE,
);
```

### Crontab

Create the file /etc/cron.d/simplesamlphp and add the following.

- Set the time (*/30 * * * *) to match the metadata attribute cacheDuration
- Set the hostname
- Set the key to match the key from config/module_cron.php
- If this is a test environment that uses a snake oil certificate add -k to curl

```
*/30 * * * * www-data curl --silent "https://host.domain.loc/simplesaml/module.php/cron/cron.php?key=RANDOM_KEY&tag=frequent&output=xhtml" > /dev
/null 2>&1
```