Installing a SimpleSAMLphp SP

SimpleSAMLphp is an alternative SP implementation that can be used in place of Shibboleth SP - and can be particularly suitable on hosted servers without root access or the ability to install full software packages. This page documents the basic install of SimpleSAMLphp Service Provider and the configuration steps necessary to integrate the SP into Tuakiri.

Full SimpleSAMLphp documentation is available at http://simplesamlphp.org/

Note that while this page uses Apache as the web server SimpleSAMLphp is deployed into, SimpleSAMLphp could be used with a number of other web servers - and it's advantage over Shibboleth SP would be the independence of Apache. Please adjust the Apache specific steps to what would work with your web server.

Prerequisites

- A web server (Apache installed) with PHP (5.2.0+)
    - To meet the PHP version requirement, the OS has to be CentOS/RHEL 6 (CentOS 5 has only PHP 5.1.x)
- The following PHP modules:
    - XML DOM (php-xml)
    - MCrypt (php-mcrypt)
    - Basic PDO database support (php-pdo) for storing sessions (at least SQLite3).
    - Optionally, also MySQL support (php-mysql)

```
yum install httpd mod_ssl php php-mcrypt php-xml php-pdo
```

- Configure SELinux: if your system has SELinux enabled, allow Apache to send email (otherwise, invocation of sendmail(postfix) from PHP breaks):

```
setsebool -P httpd_can_sendmail on
```

    - And if using SELinux, also install the `policycoreutils-python` package to get the `semanage`command which we will need later:

```
yum install policycoreutils-python
```

Basic steps

- Download simpleSAMLphp from https://simplesamlphp.org/download (1.14.8 as of August 2016)
    - Install into `/opt` and not `/var` as instructed in the SimpleSAMLphp manual.
    - In the web server space, SimpleSAMLphp will be accesible as `/simplesaml`

```
cd /opt
tar xzf ~/inst/simplesamlphp-1.14.4.tar.gz
mv simplesamlphp-1.14.4 simplesamlphp
```

- Alias this directory as `/simplesaml` - create `/etc/httpd/conf.d/simplesaml.conf` with:

```
Alias /simplesaml /opt/simplesamlphp/www
```

    - And on systems with Apache 2.4 (like CentOS or RHEL 7), explicitly grant permission to the directory - otherwise, Apache would reject to server the SimpleSAMLphp code - default access in Apache 2.4 is `Require all denied`: add this to the `/etc/httpd/conf.d/simplesaml.conf` file created above:

```
<Directory /opt/simplesamlphp/www>
  AllowOverride none
  Require all granted
</Directory>
```

- Do some basic changes to `/opt/simplesamlphp/config/config.php`:
  - Set `auth.adminpassword` to a new password.
  - Set secretsalt to a new random string (can also generate with "

    ```
    openssl rand -base64 24
    ```

    ")
  - Set technical contact name and email address.
  - Optionally, set timezone to `'Pacific/Auckland'` - or leave as `NULL` to rely on OS
- Configure a SQL session store to store sessions in a local database (even if just a sqlite3 file) instead of PHPsessions.
- Edit `config.php` and set the following:

```
        'store.type' => 'sql',
        'store.sql.dsn' => 'sqlite:/opt/simplesamlphp/data/sqlitedatabase.sq3',
```

- And give Apache write access to the "data" directory (this also means setting the SELinux context if SELinux is enabled on your system):

```
mkdir /opt/simplesamlphp/data
chown apache.apache /opt/simplesamlphp/data
# Set SELinux context to give Apache RW access - if SELinux is enabled on your system
chcon -t httpd_sys_rw_content_t /opt/simplesamlphp/data//
# And record the context setting in the SELinux policy database so that it goes not get lost in a
SELinux relabel
semanage fcontext -a -t httpd_sys_rw_content_t '/opt/simplesamlphp/data(/.*)?'
```

Configuring SP

We will be using **`sp.example.org`** to refer to the hostname of your Service Provider - please substitute that with the actual hostname of your SP.

- Create a certificate (self-signed for 20 years)

```
cd /opt/simplesamlphp/cert
openssl req -newkey rsa:2048 -new -x509 -days 7304 -nodes -out saml.crt -keyout saml.pem
```

  - ... and enter all information requested ("." to skip) - the crucial part is your hostname.
- Make the private key readable only to the user SimpleSAMLphp runs as (`apache`)

```
chown apache.apache /opt/simplesamlphp/cert/saml.{crt,pem}
chmod 600 /opt/simplesamlphp/cert/saml.pem
```

- Edit `/opt/simplesamlphp/config/authsources.php` and add references to the certificate to the default-sp definition:

```
        'default-sp' => array(
                'saml:SP',
                'privatekey' => 'saml.pem',
                'certificate' => 'saml.crt',
```

Loading the federation metadata

- Enable and configure the `metarefresh` and `cron` modules:

```
cd /opt/simplesamlphp
touch modules/metarefresh/enable
cp modules/metarefresh/config-templates/*.php config/
touch modules/cron/enable
cp modules/cron/config-templates/*.php config/
```

- Create a directory to cache the downloaded federation metadata (writable by Apache - this also means setting the SELinux context if SELinux is enabled on your system):

```
mkdir /opt/simplesamlphp/metadata/metarefresh-tuakiri
chown apache.apache /opt/simplesamlphp/metadata/metarefresh-tuakiri
# Set SELinux context to give Apache RW access - if SELinux is enabled on your system
chcon -t httpd_sys_rw_content_t /opt/simplesamlphp/metadata/metarefresh-tuakiri/
# And record the context setting in the SELinux policy database so that it goes not get lost in a
SELinux relabel
semanage fcontext -a -t httpd_sys_rw_content_t '/opt/simplesamlphp/metadata/metarefresh-tuakiri(/.*)?'
```

- Download the metadata signing certificate for the federation metadata into `/etc/shibboleth`:
  - For Tuakiri, run:

```
wget https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-cert.pem -O /opt/simplesamlphp/cert
/tuakiri-metadata-cert.pem
```

  - or for Tuakiri-TEST, run:

```
wget https://directory.test.tuakiri.ac.nz/metadata/tuakiri-test-metadata-cert.pem -O /opt
/simplesamlphp/cert/tuakiri-test-metadata-cert.pem
```

- Edit `config/config-metarefresh.php`:
  - Replace `'kalmar'` with the federation name (`'tuakiri'`)
  - Set the download URL - either for Tuakiri Production:

```
   'src' => 'https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-signed.xml',
```

  - Or Tuakiri-TEST:

```
   'src' => 'https://directory.test.tuakiri.ac.nz/metadata/tuakiri-test-metadata-signed.xml',
```

  - Set output directory and format (use `'serialize'` format):

```
                        'outputDir'     => 'metadata/metarefresh-tuakiri/',
                        'outputFormat' => 'serialize',
```

  - Set expiry date to 7 days to match Tuakiri

```
                        'expireAfter'           => 60*60*24*7, // Maximum 7 days cache time.
```

  - Change the list of accepted certificates to the metadata signing certificate downloaded above (use `tuakiri-test-metadata-cert.pem` for Tuakiri-TEST:

```
                                        'certificates' => array(
                                               'tuakiri-metadata-cert.pem',
                                        ),
```

  - Remove/comment-out the `validateFingerprint` entry (see note below for explanation)

Older versions of SimpleSAMLphp did not support directly referring to a certificate and instead required embedding the certificate fingerprint in the configuration.

For historical and archival purposes, the instructions are included here - but can be ignored in favour of using the above `certificates` setting:

- Set the 'validateFingerprint' to the fingerprint value of the metadata issuing certificate
  - Tuakiri-PROD: `06:85:C5:89:2F:38:83:98:77:1B:A4:5D:58:A4:06:3A:A4:C1:CE:45`
  - Tuakiri-TEST: `5E:90:2D:F9:D9:5A:5A:95:BF:58:4D:02:AD:29:35:64:CC:BF:76:45`
- To calculate the fignerprint yourself:
  - Download the metadata signing certificate (for Tuakiri-PROD and Tuakiri-TEST, they are linked from the instructions on registering an SP into Tuakiri)
  - and get the fingerprint value with:

```
openssl x509 -fingerprint -noout -in metadata-cert.pem
```

- Edit `config/config.php` and add an extra entry into 'metadata.sources'

```
   array('type' => 'serialize', 'directory' => 'metadata/metarefresh-tuakiri'),
```

- Now go with your browser to your SimpleSAMLphp page: https://sp.example.org/simplesaml/
  - and go to the Configuration page (log in with the Administrator password).
  - and from there to "Cron module information page"
- Edit `config/module_cron.php` and change the secret **`'key'`** from the default of `'secret'` to a different password (to prevent potential abuse of the cron URL):

```
        'key' => 'top-secret',
```

- Paste the **hourly** cron job entry (invoking curl to localhost) into root's crontab:

```
01 * * * * curl --silent "https://sp.example.org/simplesaml/module.php/cron/cron.php?
key=secret&tag=hourly" > /dev/null 2>&1
```

  - (run "crontab -e" and paste the line into the editor)

Note - invoking curl

1. If you have changed the cron password as instructed above, the line would be different than shown here.
2. If your web server is running with a self-signed HTTPS certificate, you would need to tell curl to either trust the local host certificate, or switch off certificate checking altogether.
   - Otherwise, with the `--silent` option, curl would just silently fail)
   - So use either

```
01 * * * * curl --cacert /etc/pki/tls/certs/localhost.crt --silent
"https://sp.example.org/simplesaml/module.php/cron/cron.php?
key=secret&tag=hourly" > /dev/null 2>&1
```

   - or

```
01 * * * * curl --insecure --silent "https://sp.example.org/simplesaml
/module.php/cron/cron.php?key=secret&tag=hourly" > /dev/null 2>&1
```

   - And wait for a confirmation email to be sent to the technical contact email address after the cronjob runs at HH:01.

  - You can force the job to run immediately by clicking on one of the **`hourly`** link at the bottom of the page (or pasting the cron-job URL into your browser - but the GUI link gives more output).
  - To see the output from the metadata refresh itself, go to https://sp.example.org/simplesaml/module.php/metarefresh/fetch.php

- After confirming the cron job works (wait for up to an hour to receive a confirmation email to the SimpleSAMLphp technical contact email address), edit `config/module_cron.php`and to avoid getting a confirmation email each time the cron-job runs, set
  - either **`'debug_message' => FALSE,`** (to suppress the confirmation debug message)
  - or **`'sendemail' => FALSE,`** (to suppress all email messages from the cron module)
  - However, they will have the same effect - as any error messages from metarefresh do not propagate to the cron module and are only visible in Apache error logs (`/var/log/httpd/ssl_error_log`)

Configure the SP to use the Tuakiri Discovery Service

- Edit `config/authsources.php` and set `'discoURL'`to either:
  - Tuakiri-Production: `'https://directory.tuakiri.ac.nz/ds/DS'`
  - or Tuakiri-TEST: `'https://directory.test.tuakiri.ac.nz/ds/DS'`

I.e., either

```
                'discoURL' => 'https://directory.tuakiri.ac.nz/ds/DS',
```

or

```
                'discoURL' => 'https://directory.test.tuakiri.ac.nz/ds/DS',
```

Configure Additional Tuakiri Attributes

From the list of attributes used within Tuakiri and the list of Attributes supported by SimpleSAMLphp in the default configuration, the following need to be explicitly added:

- `auEduPersonSharedToken`
- `homeOrganizationType`
- `eduPersonAssurance`

- Create `attributemap/tuakiri-attrs.php`with the following contents (adding on to what already exists in attributemap/oid2name.php)

```
<?php
$attributemap = array(
        'urn:oid:1.3.6.1.4.1.25178.1.2.10' => 'schacHomeOrganizationType',
        'urn:oid:1.3.6.1.4.1.5923.1.1.1.11' => 'eduPersonAssurance',
        'urn:oid:1.3.6.1.4.1.27856.1.2.5' => 'auEduPersonSharedToken',
);
?>
```

- Edit `config/config-metarefresh.php`and add a reference to this file in the template for IdPs downloaded via metarefresh:

```
                                        'template' => array(
                                                'tags'  => array('tuakiri'),
                                                'authproc' => array(
                                                        51 => array('class' => 'core:AttributeMap',
'oid2name', 'tuakiri-attrs'),
                                                ),
                                        ),
```

- To also configure friendly attribute names, add the following after the first line of `dictionaries/attributes.definition.json` (note that e`duPersonAssurance` already has an entry there, does not need to be duplicated)

```
  "attribute_schachomeorganizationtype": {
          "en": "Home organization type"
  },
  "attribute_auedupersonsharedtoken": {
          "en": "Shared token"
  },
```

Clean up SP configuration

Remove (comment-out) pre-configured IdPs and SPs

- Edit `metadata/saml20-idp-remote.php` - remove pre-configured `openidp.feide.no`
- Edit `metadata/saml20-sp-remote.php` - remove pre-configured `saml2sp.example.org and google.com`
- Edit `metadata/shib13-sp-remote.php` - remove pre-configured `sp.shiblab.feide.no`

Register into Federation Registry

The Tuakiri Federation Registry (FR) has in the initial setup only pre-configured support for Shibboleth SP implementation, not SimpleSAMLphp. Without the pre-configured support, it is necessary to enter all endpoints URLs manually. There is an ongoing project to add support to FR to support SimpleSAMLphp, until then, please use the Advanced Registration form as described in this section.

As a reference point, the metadata for your SP can be accessed at https://sp.example.org/simplesaml/module.php/saml/sp/metadata.php/default-sp?output=xhtml

For reference, please also see the attached image mapping SimpleSAMLphp metadata to Federation Registry form (credits: Bevan Rudge, University of Auckland).

Access the Federation Registry at the correct URL for the respective federation:

- Tuakiri (Production): https://registry.tuakiri.ac.nz/federationregistry/
- Tuakiri-TEST: https://registry.test.tuakiri.ac.nz/federationregistry/

- Start registering a new SP
- Enter your personal details
- Select your organization (Create an Organization first if not already listed)
- Enter the details about your SP (name, description, service URL)
- If SimpleSAMLphp is not listed as a supported implementation, select Advanced Registration and enter the following information (drawing from your SP metadata and using the mapping as in the image above), *replacing sp.example.org* with the hostname of your SP:

- **Entity Descriptor ID**: https://sp.example.org/simplesaml/module.php/saml/sp/metadata.php/default-sp
- **Assertion Consuming Service (Post)**: https://sp.example.org/simplesaml/module.php/saml/sp/saml2-acs.php/default-sp (Index: 0)
- **Assertion Consuming Service (Artifact)**: https://sp.example.org/simplesaml/module.php/saml/sp/saml2-acs.php/default-sp (Index: 2)

- **Single Logout Redirect Endpoint**: https://sp.example.org/simplesaml/module.php/saml/sp/saml2-logout.php/default-sp
- **Single Logout SOAP Endpoint**: https://sp.example.org/simplesaml/module.php/saml/sp/saml2-logout.php/default-sp

- **Discovery Response**: https://sp.example.org/simplesaml/module.php/saml/sp/discoresp.php

- Leave other fields blank

- Certificate: paste in the contents of cert/saml.crt

- Attributes: select the attributes needed by your SP (and give a reason for requesting each of the attributes)

- Review the SP registration form and submit it for approval.

Testing

Test authentication by going to https://sp.example.org/simplesaml/module.php/core/authenticate.php?as=default-sp

Or, to test integration with a simple application, create a PHP file with the following contents within your document space:

```
<!DOCTYPE html>
<HTML>
<PRE>
<?php

// load the SimpleSAMLphp classes
require_once('/opt/simplesamlphp/lib/_autoload.php');

// select the default authentication source
$as = new SimpleSAML_Auth_Simple('default-sp');

// require authentication
$as->requireAuth();

// get the attributes
$attributes = $as->getAttributes();

// print the attributes and the NameID
print_r($attributes);
print_r($as->getAuthData('saml:sp:NameID'));
```

```
// print out eduPersonTargetedID (which is a DOM XML NameID node as of SimpleSAMLphp 1.14)
if (isset($attributes['eduPersonTargetedID'])) {
  $eptid = $attributes['eduPersonTargetedID'][0]->item(0);
  $nameID = new SAML2_XML_saml_NameID($eptid);
  print_r($nameID);
};

?>
</PRE>
</HTML>
```

References

- Initial documentation written by Bevan Rudge: https://wiki.auckland.ac.nz/display/nesiproj/Integrating+Tuakiri+with+SimpleSAMLphp
- SimpleSAMLphp installation manual: http://simplesamlphp.org/docs/stable/simplesamlphp-install
- SimpleSAMLphp SP quick start manual: http://simplesamlphp.org/docs/stable/simplesamlphp-sp
- SimpleSAMLphp SP configuration reference: https://simplesamlphp.org/docs/stable/saml:sp
- SimpleSAMLphp documentation on integrating into a federation: http://simplesamlphp.org/docs/stable/simplesamlphp-ukaccess
- SimpleSAMLphp metadata refresh documentation: http://simplesamlphp.org/docs/stable/simplesamlphp-automated_metadata