

Integración mediante SAML 2 usando simpleSAMLphp

La integración de una aplicación PHP en el SSO usando el protocolo SAML 2 se puede realizar mediante el software **simpleSAMLphp**. Es un software publicado con licencia libre muy usado en el mundo de las federaciones de identidad basadas en SAML 2.

Al contrario que otras bibliotecas de integración, **simpleSAMLphp** requiere de una instalación básica para conseguir poner en marcha lo que se conoce como un SP (Service Provider).

Instalación

simpleSAMLphp tiene los siguientes requisitos:

- PHP 5.2.0 o superior
- Extensiones de PHP: date, dom, hash, libxml, openssl, pcre, SPL, zlib, mcrypt

Tras haberse asegurado de tenerlas todas, deberá descargar la última versión del software y descomprimirla en un directorio que no sea directamente accesible desde el servidor web. Por ejemplo:

```
# cd /var
# tar xzf simplesamlphp-1.xxxxx.tar.gz
# mv simplesamlphp-1.xxxxx simplesamlphp
```

Lo siguiente será indicarle al servidor web que debe servir cierto subdirectorio de simpleSAMLphp en una determinada ruta. Por ejemplo, en Apache lo podría hacer añadiendo la siguiente definición a la configuración de un VirtualHost servido mediante SSL:

```
<VirtualHost *:443>
    # ...
    Alias /simplesaml /var/simplesamlphp/www
    # ...
</VirtualHost>
```

Nota: Si quiere usar otra ruta distinta de /simplesaml, deberá actualizar también el parámetro baseurlpath del fichero config.php de simpleSAMLphp.

Proceda entonces a editar el fichero config/config.php de simpleSAMLphp y modifique los siguientes parámetros como se indica:

- admin.adminpassword: contraseña de administración, en claro. Será necesaria más adelante.
- admin.protectindexpage: proteger la página principal, se recomienda ponerla a true.
- secretsalt: es una cadena que servirá para la generación de elementos aleatorios. Se puede poner cualquiera, aunque se aconseja generarla de manera aleatoria con una orden como la siguiente:

```
$ tr -c -d '0123456789abcdefghijklmnopqrstuvwxyz' </dev/urandom | \
dd bs=32 count=1 2>/dev/null; echo
```

- technicalcontact_name y technicalcontact_email: datos de contacto técnico.
- timezone: Europe/Madrid
- logging.handler: se puede poner file para que simpleSAMLphp genere ficheros de log en el subdirectorio log/. Hay que dar los permisos correctos al directorio.
- language.default: lenguaje por defecto.

Tras la configuración básica, puede probar a dirigirse a <https://su.host/simplesaml/>. Si los pasos anteriores se han seguido correctamente, una página le solicitará su contraseña de administración (admin.adminpassword). Revise la pestaña Configuración para confirmar que todas las dependencias necesarias están cubiertas:

Verificación de su instalación de PHP

✓	Necesario	PHP Version >= 5.2. You run: 5.3.3
✓	Necesario	Hashing function
✓	Necesario	ZLib
✓	Necesario	OpenSSL
✓	Necesario	SimpleXML
✓	Necesario	XML DOM
✓	Necesario	RegEx support
✓	Necesario	MCrypt
✓	Opcional	MySQL support
✓	Necesario para LDAP	LDAP Extension
✓	Recomendado	technicalcontact_email option set
✓	Necesario	auth.adminpassword option set
✓	Recomendado	Magic Quotes should be turned off

Configuración del SP

simpleSAMLphp está funcionando, pero no sabe cuál es su función. Le indicaremos que es un SP editando el fichero `config/authsources.php` y definiendo los siguientes parámetros para la entrada `default-sp`:

```
<?php
// ...
'default-sp' => array(
    'saml: SP' ,
    'certificate' => 'mi_aplicacion.pem' ,
    'privatekey' => 'mi_aplicacion.key' ,
    'entityID' => ' https://su.host/simplesaml/' ,
    'idp' => ' https://idp.uhu.es/idp' ,
    'discoURL' => NULL,
    'redirect.sign' => TRUE,
    'redirect.validate' => TRUE,
    'assertion.encryption' => TRUE
),
```

Con esta configuración, simpleSAMLphp sabe que:

- Es un SP.
- Tiene que usar internamente los certificados *mi_aplicacion.pem* y *mi_aplicacion.key*.
- Su identificador interno es <https://su.host/simplesaml/>. En SAML es muy común usar como identificador de un SP/IdP una URL que identifique al servicio.
- El proveedor de identidad que debe usar es *idp.uhu.es*.
- Me mejora la seguridad en las comunicaciones encriptándolas y validándolas.

Certificados

El par de claves que usaremos como certificado puede ser autogenerado, no es necesario que esté emitido por ninguna CA de confianza, ni que sea el mismo certificado usado por el servidor web de la aplicación.

Generarlo es bastante sencillo:

```
# cd cert/
# openssl req -newkey rsa:2048 -new -x509 -days 3652 -nodes \
    -out mi_aplicacion.pem -keyout mi_aplicacion.key
Generating a 2048 bit RSA private key
.....+++
```

```
.....+++
writing new private key to 'mi_aplicacion.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:ES
State or Province Name (full name) []:Sevilla
Locality Name (eg, city) [Default City]:Sevilla
Organization Name (eg, company) [Default Company Ltd]:Universidad de Sevilla
Organizational Unit Name (eg, section) []:Mi grupo
Common Name (eg, your name or your server's hostname) []:su.host
Email Address []:sucorreo@us.es
```

Metadatos del SSO (IdP)

Los certificados están listos, pero el SP aún no sabe quién es idp.uhu.es. Para que lo sepa hay que proporcionarle los metadatos del SSO como IdP SAML 2.0. Para ello primero tenemos que activar el módulo de `cron`:

```
# cd /var/simplesaml/modules/cron
# touch enable
# cp config-templates/module_cron.php ../../config
```

En el fichero de configuración `/var/simplesamlphp/config/module_cron.php` ponemos lo siguiente:

```
$config = array (
    'key' => 'laclavequesequiera',
    'allowed_tags' => array('daily', 'hourly', 'frequent'),
    'debug_message' => FALSE,
    'sendemail' => FALSE,
);
```

Si todo ha ido bien podremos acceder a la dirección

<https://su.host/simplesaml/module.php/cron/croninfo.php> en donde se nos indicará lo que tendremos que poner en el `cron` del sistema. Por lo general bastará ejecutar lo siguiente:

```
#crontab -e

# Ejecutar cron: [daily]
```

```
02 0 * * * curl --silent "https://su.host/simplesaml/module.php/cron/cron.php?key=lkfEp0UwPB&tag
# Ejecutar cron: [hourly]
01 * * * * curl --silent "https://su.host/simplesaml/module.php/cron/cron.php?key=lkfEp0UwPB&tag
# Ejecutar cron: [frequent]
5,15,25,35,45,55 * * * * curl --silent "https://su.host/simplesaml/module.php/cron/cron.php?key=
```

Si los certificados usados en apache son autofirmados o no se pueden verificar, añadiremos la opción `[-insecure]` al comando `[curl]`.

El siguiente módulo a activar es el `[metarefresh]`:

```
# cd /var/simplesaml/modules/metarefresh
# touch enable
# cp config-templates/config-metarefresh.php ../../config
```

Editamos la configuración de este módulo localizada en `/var/simplesamlphp/config/config-metarefresh.php` y ponemos:

```
$config = array(
    'sets' => array(
        'uhu' => array(
            'cron' => array('daily'),
            'sources' => array(
                array(
                    'src' => 'https://janus.uhu.es/idp/module.php/janus/meta
                    'validateFingerprint' => '934877BF983789B34ABA45CE3DD484
                    'template' => array(
                        'redirect.sign' => true,
                        'redirect.validate' => true,
                        'tags' => array('uhu'),
                        'assertion.encryption' => true,
                    ),
                ),
            ),
        ),
        'expireAfter' => 60*60*24*2, // Maximum 2 days cache time.
        'outputDir' => 'metadata/uhu/',
        'outputFormat' => 'flatfile',
    ),
),
)
```

Además de esto en el fichero de configuración de simplesamlphp `/var/simplesamlphp/config/config.php` hay que añadir lo siguiente:

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

Para que funcione correctamente nos tenemos que asegurar que el usuario de apache tiene permisos de escritura en el directorio `/var/simplesamlphp/federation`.

Solicitud de alta

El siguiente paso consistirá en solicitar el acceso al SSO. Para ello primero tiene que obtener los metadatos de su SP. El recuadro de metadatos que encontrará debajo debe contener los metadatos de su SP. Para obtenerlos, acceda a <https://su.host/simplesaml>, haga click en la pestaña Federación y busque el siguiente enlace:

Integración de la aplicación

Metadatos SP SAML 2.0

Entity ID: <https://hdvirtual.us.es/simplesaml/>

simpleSAMLphp proporciona una interfaz muy simple para que cualquier aplicación en PHP haga uso de su funcionalidad. Puede ver la documentación completa de integración en [SP API reference](#).

Para integrar su aplicación, necesita cargar las bibliotecas de simpleSAMLphp:

```
<?php
require_once ' /var/simplesamlphp/lib/_autoload.php';

$as = new SimpleSAML_Auth_Simple(' default-sp' );
<code>
\\
Para forzar la autenticación y leer los atributos cuando el usuario ya esté autenticado, puede u
<code>
<?php
// Carga de biblioteca

$as->requireAuth();

$attributes = $as->getAttributes();
```

Hay más métodos disponibles. Puede verlos en [SP API reference](#).

Revisión #1

Creado 18 noviembre 2021 16:37:26 por Tecnicos Aulas externo

Actualizado 18 noviembre 2021 16:41:53 por Tecnicos Aulas externo