Configuring a simpleSAMLphp SP for SURFsecureID

The sp.php script below shows how you can use SimpleSAMLphp to:

- Request a specific minimum level op assurance (LoA) from the SURFsecureID gateway
- Verify the LoA at which the user is authenticated

To use this scripts:

1. Configure a SimpleSAMLphp SAML 2.0 hosted SP with the name "default-sp". You can following the instructions on the simpleSAMLphp website.
2. Add the SURFsecureID gateway as SAML 2.0 identity provider to `<simplesaml>/metadata/saml20-idp-remote.php`. Below you find a `saml20-idp-remote.php` containing the metadata of the SURFsecureID environments in simpleSAMLphp format for your convenience.
3. Put the sp.php script in the <simplesaml>/www directory. Uncomment the `$gLOAmap` and `$gRemoteIDPEntityID` variables for the SURFsecureID environment that you are connecting to.

sp.php

```
<?php
/*
Copyright 2014, 2019 SURFnet bv

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*/

// Include SimpleSAMLphp. Assume this script is placed in the <simplesaml>/www dir.
require_once('../lib/_autoload.php');

// Name of session variable for storing the minimum required LOA for a login
define( 'SSP_SESSION_MIN_LOA', 'RequestedMinLOA' );

/* Build return URL. This is where to ask simplesamlPHP to direct the browser to after login or logout
   Point to this script, but without any request parameters so we won't trigger an login again (and again,
and again, and ...)
   Note: If your SP is located behind a HTTP proxy, you may need to update the way that this URL is
generated, depending on the
         configuration of the proxy.
*/
$returnURL = ($_SERVER['HTTPS'] == 'on') ? 'https://' : 'http://';
$returnURL .= $_SERVER['HTTP_HOST'];
$returnURL .= $_SERVER['SCRIPT_NAME'];

/* $gLOAmap map integer level of assurance level to the identifier used by SURFsecureID
   $gSURFsecureIDIdPEntityID is set to the EntityID of the SURFsecureID IdP
   Note: The LoA and IdP EntityID identifiers are different for each SURFsecureID environment. Uncomment the
identifiers for the
         environment that you are using. See: https://wiki.surfnet.nl/display/SsID
/SURFsecureID+Metadata+for+Service+Providers
*/

/*
// SURFsecureID Production environment
// EntityID
$gSURFsecureIDIdPEntityID = 'https://sa-gw.surfconext.nl/authentication/metadata';
// LoA identifiers
$gLOAmap = array(
    1 => 'http://surfconext.nl/assurance/loa1',
    2 => 'http://surfconext.nl/assurance/loa2',
    3 => 'http://surfconext.nl/assurance/loa3',
);
*/
```

```
// SURFsecure Test environment
// EntityID
$gSURFsecureIDIdPEntityID = 'https://sa-gw.test.surfconext.nl/authentication/metadata';
// LoA identifiers
$gLOAmap = array(
    1 => 'http://test.surfconext.nl/assurance/loa1',
    2 => 'http://test.surfconext.nl/assurance/loa2',
    3 => 'http://test.surfconext.nl/assurance/loa3',
);

try {
    // Init SP instance
    // Assumes you have setup a SP named "default-sp" in <simplesaml>/config/authsources.php
    // See: https://simplesamlphp.org/docs/stable/simplesamlphp-sp
    $as = new SimpleSAML_Auth_Simple('default-sp');    // Init SP instance

    /** @var $session SimpleSAML_Session */
    $session = SimpleSAML_Session::getSessionFromRequest();

    // Process login action. Assumes the login function of your SP uses ...?action=login
    if (isset($_REQUEST['action']) && $_REQUEST['action'] == 'login' ) {

        // We use the SSP session to keep track of the LOA we want.
        // Unset any existing RequiredAuthnContextClassRef
        $session->deleteData('string', SSP_SESSION_MIN_LOA);

        // login
        $requiredLOA = 2; // The LOA level we want.

        // Store the requested LOA in the session so we can verify it later
        $session->setData('string', SSP_SESSION_MIN_LOA, $requiredLOA);

        $as->login( array(
            'ReturnTo' => $returnURL,
            'ForceAuthn' => false,
            'saml:AuthnContextClassRef' => $gLOAmap[$requiredLOA]  // Specify LOA
        ) );

        exit;   // Never reached. Added for clarity
    }

    // Process logout action
    if( isset($_REQUEST['action']) && $_REQUEST['action'] == 'logout' ) {
        $as->logout( array (
            'ReturnTo' => $returnURL,
        ) );  // Process logout

        exit;   // Never reached. Added for clarity
    }

    // Output HTML page

    echo <<<head
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.
dtd">
    <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
        <head>
            <meta http-equiv="Content-type" content="text/html;charset=UTF-8" />
            <style type="text/css">
                table,th,td {border: 1px solid black;}
                th,td {padding 1px}
            </style>
            <title>simpleSAMLphp LoA Authorisation Demo</title>
        </head>
        <body>
            <h1>SimpleSAMLphp LoA Authorisation Demo</h1>
head;

    // Page to show when SAML authentication was successful
    if ( $as->isAuthenticated() ) {
        // Get attributes from SAML Assertion
        $attributes = $as->getAttributes();

        // Read the LoA we wanted from the SSP session
```

```
        $requestedLoA = $session->getData('string', SSP_SESSION_MIN_LOA);    // What we requested during
login

        $idpEntityID = $as->getAuthData('saml:sp:IdP');
        $nameID = $as->getAuthData('saml:sp:NameID');

        // Get the authentication state
        $authState = $session->getAuthState('default-sp');
        $authnConext = $authState['saml:sp:AuthnContext'];
        $authnInstant = gmdate('r', $authState['AuthnInstant'] );
        $expire = gmdate('r', $authState['Expire'] );

        echo "<h2>You are logged in</h2>";

        echo "<h3>SimpleSAMLphp Session</h3>";
        echo "<p>SimpleSAMLphp session start: <code>{$authnInstant}</code></br />";
        echo "SimpleSAMLphp session expire: <code>{$expire}</code></p>";

        echo "<h3>LoA and Authorisation</h3>";
        $authnConextHTML=htmlentities($authnConext);
        $IdPEntityIDHTML=htmlentities($idpEntityID);
        echo "<p>Authenticated by IdP with EntityID <code>{$IdPEntityIDHTML}</code><br />";
        if ( strlen($authnConext) > 0 ) {
            echo "<p>Received authnConext <code>{$authnConextHTML}</code></p>";
        }
        else {
            echo "<p>The IdP did not send an authnConext</p>";
        }

        // We still need to verify that the Authentication is at the LoA we requested, and that the
authenticaton
        // actually came from the IdP we expected. Depending on the way SimpleSAMLphp is configured multiple
IdPs
        // could be allowed to authenticate to the SP. When using the LoA level for authorization it is
important
        // to ensure that the LoA identifier is comming from an IdP that we trust to issue this identifier.

        // Map LoA identifier back to an integer LoA level
        $actualLoA = -1;
        $translatedLoA = array_search($authnConext, $gLOAmap);
        if (false ==! $translatedLoA) {
            $actualLoA = $translatedLoA;
            echo "<p>Actual LoA is: <b>{$actualLoA}</b></p>";
        }
        else {
            echo "<p>Unrecognised LoA identifier: <code>$authnConextHTML</code></p>";
            $actualLoA = -1; // Unrecognised LoA identifier
        }

        // Verify that the IdP that authenticated is the one we expected
        if ( $idpEntityID !== $gSURFsecureIDIdPEntityID ) {
            echo "<p>Authenticated by untrusted IdP: <code>$idpEntityID</code></p>";
            $actualLoA = -1; // Wrong IdP
        }

        if (NULL !== $requestedLoA) {
            echo "<p>Requested LoA was: <b>{$requestedLoA}</b></p>";
            if ($actualLoA >= $requestedLoA) {
                echo '<p><b>Success!</b> You were authenticated at or above the minimally required LoA</p>';
            }
            else {
                echo '<p><b>Failed!</b> You were not authenticated at the required LoA</p>';
                // TODO: Handle this authorisation failure
            }
        }

        $nameIDValueHTML=htmlentities($nameID['Value']);
        $nameIDFormatHTML=htmlentities($nameID['Format']);

        echo <<<html
        <h3>NameID</h3>
        <table>
            <tr><th>Value</th><td><code>{$nameIDValueHTML}</code></td></tr>
            <tr><th>Format</th><td><code>{$nameIDFormatHTML}</code></td></tr>
        </table>
```

```
html;
        echo <<<html
        <h3>SAML Attributes</h3>
        <table>
            <tr><th>Attribute</th><th>Value(s)</th></tr>
html;
        foreach ($attributes as $attrName => $attrVals) {
            $attrNameHTML=htmlentities($attrName);
            echo "<tr><td><code>{$attrNameHTML}</code></td><td>";
            foreach ($attrVals as $attrVal) {
                $attrValHTML=htmlentities($attrVal);
                echo "<code>{$attrValHTML}</code><br />";
            }
            echo "</td>";
        }
        echo <<<html
        </table>

        <h3>Logout</h3>
        <p>
            <form name="logout" action="{$returnURL}" method="get">
               <input type="hidden" name="action" value="logout"/>
               <input type="submit" value="Logout" />
            </form>
        </p>
html;
    }

    // User is not authenticated
    else {
        echo <<<html
            <h2>Your are not logged in</h2>
html;
    }

    echo <<<html
            <h3>Login (again)</h3>
            <p>
                <form name="login" action="{$returnURL}" method="get">
                   <input type="hidden" name="action" value="login"/>
                   <input type="submit" value="Login" />
                </form>
            </p>
html;

    echo <<<html
        </body>
    </html>
html;
}
catch (Exception $e)
{
    echo $e->getFile().':'.$e->getLine().' : '.$e->getMessage();
}
```

saml20-idp-remote.php

```
<?php
// SURFsecureID PRODUCTION environment
$metadata['https://sa-gw.surfconext.nl/authentication/metadata'] = array (
    'entityid' => 'https://sa-gw.surfconext.nl/authentication/metadata',
    'metadata-set' => 'saml20-idp-remote',
    'sign.authnrequest' => true,
    'signature.algorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha256',
    'SingleSignOnService' =>
        array (
            0 =>
                array (
                    'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
                    'Location' => 'https://sa-gw.surfconext.nl/authentication/single-sign-on',
                ),
        ),
    'keys' =>
        array (
            0 =>
                array (
                    'encryption' => false,
                    'signing' => true,
                    'type' => 'X509Certificate',
                    'X509Certificate' =>
'MIICsjCCAZoCCQDHN3+HzElEDDANBgkqhkiG9w0BAQUFADAbMRkwFwYDVQQDFBBnYXRld2F5X3NhbWxfaWRwMB4XDTE1MDcyMzEyMTUxOVoX
DTIwMDcyMTEyMTUxOVowGzEZMBcGA1UEAxQQZ2F0ZXdheV9zYW1sX2lkcDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALK
/JwHWd5JftXYKO9qcTQ4dfKEnl35oJj6PlEyR6gpikdpgm2OY/zy4e7vcXfBChedVF3OUI4rRDWCz4yXT2sldzjuIyONJfA86xva5lxDARqT
/+gRBuZ2pyMTb0okvl1G9ZlAjPumVH14591rp6OGT5TJIkILQ
/pKp1INdiBqpiR53Z5YvsXEUJ8PHHZyILO00HnBldq0d77lmATr6QamXpbY+CZ9pIw65t32fhFcUfRC68C81/P2crCn3v5GMyrQcM/tB/xdVf
/haEZiqgI/bjcreBpQobnAhwEsve+uvbSLFN1Rsc7o0W
/7Pn6EGBX1h9rjKjDgqssHuWkVuU4sCAwEAATANBgkqhkiG9w0BAQUFAAOCAQEAmlqfTvEfGDeqqqvuAMDG5IKDo6h21wwByywNbRhimfOvL6
FqIgAgx+D3gxW1lO41PcqQQKYIVUEAuYv+tW8COLdHcFRh/UV9ei4iquMwBCkO
/XOoMC9FsRBo3yPaQClRK8OYj1IXer4JXNuFHeLblzf+GLYFoqMWWwT2dnBLAePoEgANKUm2aasxyiJmnroNa+O5zTP9ExT3qHphCCG1gh3iH
rQu9iSEJxY12zAQYtPomIs8Vk/GBfj+ucUiBEEqaMpCH+t6f0VOIoP1SNHgNAaeBLVuOpS0VlLnwZFJkNPVOQpFgRuoFsH3
/9i53Fs3eQreb9wzq2VkjDhhlc5eyA==',
                ),
        ),
);

// SURFsecureID TEST environment
$metadata['https://sa-gw.test.surfconext.nl/authentication/metadata'] = array (
    'entityid' => 'https://sa-gw.test.surfconext.nl/authentication/metadata',
    'metadata-set' => 'saml20-idp-remote',
    'sign.authnrequest' => true,
    'signature.algorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha256',
    'SingleSignOnService' =>
        array (
            0 =>
                array (
                    'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
                    'Location' => 'https://sa-gw.test.surfconext.nl/authentication/single-sign-on',
                ),
        ),
    'keys' =>
        array (
            0 =>
                array (
                    'encryption' => false,
                    'signing' => true,
                    'type' => 'X509Certificate',
                    'X509Certificate' =>
'MIIC6jCCAdICCQCWUmXBRox3ZDANBgkqhkiG9w0BAQsFADA3MRkwFwYDVQQDDBBHYXRld2F5IFNBTUwgSWRQMRowGAYDVQQKDBFTVVJGc2Vj
dXJlSUQgVEVTVDAeFw0xODA4MDEwODA2MjdaFw0yMzA3MzEwODA2MjdaMDcxGTAXBgNVBAMMEEdhdGV3YXkgU0FNTCBJZFAxGjAYBgNVBAoME
VNVUkZzZWN1cmVJRCBURVNUMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3okLxR7J2re6j7
/rLjEYLc7iWiELcypmFLvL9BmCYqYZ80Pn+SR9LPUOXcplFUt6LYh/NOK5JMT0P6o0OTUP1P4zEMzLEl0wSJ1jBcu88yNppJoUn
/TEgXMGNB1DW8jlVvzcgNSsJjxuw2Fj6J
/6D+b77+7PhNMagbnfBEFStz0RBv7JOBdzuEC71wVxlGXB7C1Y8ZF3AwZgIp0jOVdiMub9i6neaKV9ZBLSv+azkT8BtAauMdKBpBxC+KxUFV9
ccHKFnF2YOTLQ3CJNNGyQTtCJVI82fggraKnl+by2elV3+Dmzc0iqMcAdECasSS+2E8iOqw+3Qss+RgtS7QvCdQIDAQABMA0GCSqGSIb3DQEB
CwUAA4IBAQCq/j+uXLvYDHhL7c/Y3+oj25+ur2UtZ/uSBqZIIqGlAzlCEL/zdgDI8XmePaRLtc2hYWUH4bD5Iu8HqxrMPrdBkG
/5cjbMmlhU5uV3EX7S+m89k9vrok9+7B+uynCkMIdA/1Uif2btfEQi9hevvyP/1vvyoHqftym+ivIOyvELJNIgdTUaqvcJy
//QvkmpvSpgTvlzHSVgKkSmMoBhTmevu7lQUGYSk/Mt53Zd3WmZhev+emS/MTKwV39JkZg7aykIRqXGVe/yTlttW
/zaV9WtSIzNZfaKqASraAaClKgv8lsTjWFv88HZrsP/UuEseIWh4NjOo5HHvHYgqN/atX3t',
                ),
        ),
);
```