

CESNET Technical Report 6/2010

Running the Service Provider

PETR GROLMUS, IVAN NOVAKOV

Received 27 May 2010

Abstract

Shibboleth is a software package produced by the Internet2¹ Consortium to enable Single Sign-on (SSO) authentication for users in an environment composed of multiple organizations. Splitting Shibboleth into two parts – the Identity Provider (IdP) and the Service Provider (SP) – allows for separation of responsibilities and their assignment to individual participants. Home IdPs collect information (attributes) on their own users, provide authentication services for those users, and send pre-determined sets of attributes to the SP. As indicated by its title, an SP provides a service; authorized access to that service is allowed to users based on attributes received from IdPs. Any Web application can be seen as an example of such a service. This Report explains how to make service providers run in Linux.

Keywords: Shibboleth, Identity Provider, IdP, Service Provider, SP, Single Sign-On, SSO, authentication, authorization

1 Running the Service Provider

This section is going to explain how to run a Service Provider (SP) in Debian Linux². The operating system should have the Apache web server installed, since the Shibboleth Native SP is being developed specifically for that product:

```
aptitude install apache2-mpm-prefork
```

It is also necessary to get a suitable web server certificate from your certification authority, and configure the server to provide services over secure HTTP (https). Getting and installing a certificate is outside the scope of this Report but an experienced web server administrator can be reasonably expected to handle the installation without difficulty. Certificates can be used to establish secure connections between users and web servers as Shibboleth Service Providers with Identity Providers – see below. For the sake of completeness, the following excerpt of the Apache configuration file shows how to indicate the location of the certificate file within the filesystem:

```
SSLEngine on
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLVerifyDepth 3
SSLCertificateFile /etc/apache2/ssl/server.crt
SSLCertificateKeyFile /etc/apache2/ssl/server.key
SSLCertificateChainFile /etc/apache2/ssl/ca-chain.pem
```

¹ <http://www.internet2.edu/>

² Using the current *stable* version – Lenny 5.0.4

Developers have made the installation of Shibboleth Service Providers quite easy, since the Shibboleth Native SP package is available for the Debian Linux Distribution. Rather than the standard repositories, though, the package is only available from the *backports* repository, which means that a new source must be added to your */etc/apt/sources.list* configuration file:

```
deb http://www.backports.org/debian lenny-backports main contrib non-free
deb-src http://www.backports.org/debian lenny-backports main contrib non-free
```

It is also advisable to install GPG keys for the *backports* repository

```
aptitude install debian-backports-keyring
```

and refresh the list of packages provided by package repositories, including the new ones:

```
aptitude update
```

The Shibboleth Native SP package can then be installed by calling

```
aptitude -t lenny-backports install libapache2-mod-shib2
```

It is further necessary to activate the SP module in Apache. That is easily achieved by calling

```
a2enmod shib2
```

With the installation finished, the SP is still not fully functional and it must be configured. The following text will explain SP configuration within eduId.cz, the Czech academical identity federation.

The main configuration file for Shibboleth SP is */etc/shibboleth/shibboleth.xml*. Backing up the original file before modification is recommended. There are several changes required. Firstly, the name of the machine running the SP must be set. We are going to use *shib-sp.zcu.cz* as an example. Examples of individual settings will be shown in the context of larger sections of the XML configuration file.

The two commands below will create a backup copy of the original configuration file (titled *shibboleth2.xml-orig*) and produce a new configuration file with the name of the server replaced correctly in all locations.

```
cp shibboleth2.xml shibboleth2.xml-orig
sed "s/sp.example.org/shib-sp.zcu.cz/g" \
    shibboleth2.xml-orig > shibboleth2.xml
```

Defining the host and the protected (*secret*) directory within the *RequestMapper* element:

```
<Host name="shib-sp.zcu.cz" authType="shibboleth"
    requireSession="true">
  <Path name="secret" requireSession="true"/>
</Host>
```

Then, we need to specify the default application in the *ApplicationDefaults* element:

```
<ApplicationDefaults
  id="default" policyId="default"
  entityID="https://shib-sp.zcu.cz/shibboleth"
  homeURL="https://shib-sp.zcu.cz/secret"
  REMOTE_USER="eppn persistent-id targeted-id"
  signing="true" encryption="true">
```

Among attributes shown here, *entityID* is worth noticing as it provides a unique identification of the given Shibboleth SP within the whole eduId.cz federation. *REMOTE_USER* specifies IdP attributes that can be used to identify users. For example, the first *eppn* is *EduPersonPrincipalName* which, by agreement, contains the user's login name followed by the '@' character and the user's home organization's domain (for instance "*indy@zcu.cz*").

Next we need to define a Shibboleth SP session in the *Sessions* element:

```
<Sessions
  lifetime="28800" timeout="3600"
  checkAddress="false" consistentAddress="true"
  handlerURL="/Shibboleth.sso" handlerSSL="true"
  cookieProps="; path=/; secret"
  exportLocation="http://shib-sp.zcu.cz/Shibboleth.sso/GetAssertion"
  idpHistory="false" idpHistoryDays="7">
```

The *Sessions* element must contain at least one *SessionInitiator* defining ways of initiating sessions. Our solution relies in the WAYF (Where Are You From) service provided by the eduId.cz federation:

```
<SessionInitiator type="Chaining" Location="/DS"
  isDefault="true" id="DS" relayState="cookie">
  <SessionInitiator type="SAML2" defaultACSIndex="1"
    template="bindingTemplate.html"/>
  <SessionInitiator type="Shib1" defaultACSIndex="5"/>
  <SessionInitiator type="SAMLDS" URL="https://www.eduid.cz/wayf/" />
</SessionInitiator>
```

To simplify testing of SP settings, a minor modification of *handlers* in the *Sessions* element is also recommended. Contrary to the default settings, we remove localhost access constraints ('acl="127.0.0.1"') from "/Status" and allow display of session attribute values in the "/Session" handler:

```
<Handler type="Status" Location="/Status" />
<Handler type="Session" Location="/Session"
  showAttributeValues="true" />
```

Once proper functioning of the SP has been confirmed, these two *handlers* can (and probably should) be reset to their original values; that will cause no negative impact on the operation of the SP.

The *MetadataProvider* element provides a definition of federation metadata file downloads.

```
<MetadataProvider type="Chaining">
  <MetadataProvider
    type="XML"
    uri="http://www.eduid.cz/docs/eduid/metadata/eduid-metadata.xml"
    backingFilePath="/var/run/shibboleth/backup_eduid-metadata.xml"
    reloadInterval="1800">
  </MetadataProvider>
</MetadataProvider>
```

We are now done with the main configuration file (*shibboleth2.xml*), although the configuration process itself is not yet fully completed. We still need to set up secure communication within the federation, connect the SP into the federation, force *shibboleth-based* authentication to services, and test the settings.

Existing keys and certificates used in secure client-server transfers can also be used to establish secure communication within the federation (see the section on Apache server settings). The default key and certificate provided by the installation package (located by default in */etc/shibboleth/*) may be replaced with new ones. The default names of those files are *sp-key.pem* and *sp-cert.pem*:

```
cp /etc/apache2/ssl/server.key /etc/shibboleth/sp-key.pem
cp /etc/apache2/ssl/server.crt /etc/shibboleth/sp-cert.pem
```

Shibboleth Native SP package developers have even provided a *shibd* startup script for the SP service, and a user identity (“_shibd”) used to run the service. Unless user “_shibd” possesses sufficient rights to read the key and certificate, though, the service starts under root, which is not acceptable. It is strongly recommended to grant the “_shibd” account sufficient permissions to read the key and certificate files:

```
chown _shibd /etc/shibboleth/sp-key.pem
chown _shibd /etc/shibboleth/sp-cert.pem
```

Now, we need to set up the required IdP attributes. This is done through the */etc/shibboleth/attribute-map.xml* configuration file. In essence, the only thing that needs to be done here is uncommenting attributes required by the given application. In our case, for instance, we wanted to use applications that require the first name, last name and e-mail address of any user trying to access the application. That was why the following attributes had to be uncommented:

```
<Attribute name="urn:oid:2.5.4.3" id="cn"/>
<Attribute name="urn:oid:2.5.4.4" id="sn"/>
<Attribute name="urn:oid:2.5.4.42" id="givenName"/>
<Attribute name="urn:oid:0.9.2342.19200300.100.1.3" id="mail"/>
```

The final step consists in pointing the Apache Web Server to an address that needs to be protected by *Shibboleth*. As an example, we are going to stick with the previously used *secret* directory, and add the following section to Apache configuration:

```
<Location /secret/>
  AuthType shibboleth
  require shibboleth
</Location>
```

The startup script for the SP service is located at the usual place (*/etc/init.d/*). The service must be restarted after any configuration change in the service by calling:

```
/etc/init.d/shibd restart
```

To achieve full functionality, webserver restart must follow:

```
/etc/init.d/apache2 restart
```

Correct configuration may be verified by accessing

<https://shib-sp.zcu.cz/Shibboleth.sso/Status>

which yields an XML file showing, aside of essential directives and certificates, a section on feature configuration:

```
<Status>
  <OK/>
</Status>
```

1.1 Integrating SP with the eduId.cz Federation

To integrate the new SP into the eduId.cz Federation, SP's information must be incorporated into the Federation's central metadata. Let us assume that the given organization is already – at the administrative level – a member of the Federation with a duly appointed contact person responsible, among others, for maintaining the organization's relevant records in central metadata.

Here, once again, our job was made easy by Shibboleth Native SP developers, since appropriate metadata can easily be downloaded from

<https://shib-sp.zcu.cz/Shibboleth.sso/Metadata>

It is recommended to add additional information on the organization and contact person just before the closing tag `</md:EntityDescriptor>`, e.g.:

```
<md:Organization>
  <md:OrganizationName xml:lang="en">
    University of West Bohemia
  </md:OrganizationName>
  <md:OrganizationDisplayName xml:lang="en">
    University of West Bohemia
  </md:OrganizationDisplayName>
  <md:OrganizationURL xml:lang="en">
    http://www.zcu.cz/
  </md:OrganizationURL>
</md:Organization>
```

```
<md:ContactPerson contactType="technical">
  <md:GivenName>Petr</md:GivenName>
  <md:SurName>Grolmus</md:SurName>
  <md:EmailAddress>indy@civ.zcu.cz</md:EmailAddress>
</md:ContactPerson>
```

Once complete, metadata can be sent to *eduid-admin@eduid.cz*. The message must be electronically signed by the responsible technical contact using a personal certificate issued by CESNET. Metadata whose sender could not be verified will be ignored.

1.2 Verifying SP Functionality

We have already done basic verification by checking SP status at

https://shib-sp.zcu.cz/Shibboleth.sso/Status

Now, we need to check if the SP really works. Let us assume that relevant metadata have already been published in the Federation's central metadata repository.

So far, we have set up shibboleth-based authentication for resources located in the */secret/* directory. To check if the SP works, use a script that displays values of variables upon successful authorization. This requires the server to support the PHP scripting language. If not available, install it first by calling:

```
aptitude install libapache2-mod-php5
```

Then go to the */secret/* directory, the webserver's DocumentRoot, and create file *index.php* with the following contents:

```
<? phpinfo(); ?>
```

Accessing address *https://shib-sp.zcu.cz/secret/* should take us through authorization by the Federation's WAYF service, calling on our home organization to provide authorization. In case everything (i.e. the SP settings and required attributes) has been set up correctly, the list generated by *phpinfo()* should also include records similar to those shown below:

_SERVER["Shib-Application-ID"]	default
_SERVER["Shib-Session-ID"]	_e2ddced58220f6779f07ac1a1c6c7cda
_SERVER["Shib-Identity-Provider"]	https://shib.zcu.cz/idp/shibboleth
_SERVER["Shib-Authentication-Instant"]	2010-05-27T10:37:29.686Z
_SERVER["Shib-AuthnContext-Decl"]	urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified
_SERVER["cn"]	Ing. Petr GROLMUS
_SERVER["eppn"]	indy@zcu.cz
_SERVER["givenName"]	Petr
_SERVER["mail"]	indy@civ.zcu.cz
_SERVER["sn"]	GROLMUS

Figure 1. Attributes provided by Shibboleth

1.3 References

This Technical Report is mostly based on personal experience but also uses information published on the eduId.cz website³ of the Czech Academic Identity Federation, and on-line documentation for the Shibboleth project⁴.

³ <http://www.eduid.cz/>

⁴ <http://shibboleth.internet2.edu/>