

CESNET Technical Report 22/2009

Shibboleth authentication for Adobe Connect Pro

IVAN NOVAKOV

Received 16.12.2009

Abstract

This technical report describes the technical process of implementing Shibboleth authentication for the Adobe Connect Pro application. It is designated for system administrators with practical experience with the Shibboleth Service Provider software.

Keywords: Shibboleth idp, cluster, high availability, load balancing, terracotta

1 Introduction

Adobe Connect Pro¹ is a popular web based conferencing software. It requires only a web browser with flash support. Connected users can collaborate on different tasks, chat to each other, communicate through audio and video, share their desktops etc.

Shibboleth² is an open source implementation of federated identity based authentication and authorization infrastructure based on SAML (Security Assertion Mark-up Language).

This technical report describes how to set up external Shibboleth authentication for Adobe Connect Pro, thus making it federation-enabled. Federated Adobe Connect Pro provides easy and convenient access for users from different organizations.

2 Adobe Connect Pro Authentication Overview

Adobe Connect Pro has its own user database. To access the application the user has to enter his username and password. Adobe Connect does not support any other authentication back-ends (LDAP, SQL database, etc.) except its own user database. There is support for external authentication though, but it is somewhat simple and limited.

It is possible to configure Adobe Connect to accept the identity of the user contained in a special HTTP header. It is an extremely simple way of providing external authentication, but at the same time there may appear many related problems that need to be resolved. For example, that set-up requires a reverse proxy to work. It is important to make sure, that all connections to the Adobe Connect server go via the proxy and the server itself is not accessible in any other way. Otherwise, anyone could fake his identity by crafting the appropriate HTTP header.

¹ <http://www.adobe.com/products/acrobatconnectpro/>

² <http://shibboleth.internet2.edu/>

The Adobe Connect server has an very powerful remote XML based API [2]. Practically every operation available at the server (and through the application) is accessible through the API, including authentication. The authentication is performed through the login remote call, where the username and the password are passed as arguments. Upon successful login at the server, the response contains a session cookie, which should be used in all subsequent calls in order to retain authenticated status.

That provides an alternative way, how to implement external authentication. The scenario is as follows - the user wishing to use Adobe Connect Pro is redirected to an external application providing custom authentication. The application authenticates the user and then uses the remote login API call to the Adobe Connect server on behalf of the user. The user is then redirected to Adobe Connect Pro and the session cookie received from the login API call is passed as a GET parameter.

3 The AC Login Service Overview

AC Login Service³ is an application developed by me, which implements external Shibboleth authentication to Adobe Connect. The application is written in PHP and uses a low level client API also written in PHP to communicate with the Adobe Connect server through its remote XML API [2]. The external authentication is provided by a Shibboleth service provider instance.

The service requires a web server with PHP and SSL support to run. In this case we are using Apache 2.2 [1]. The service also uses parts of the Zend Framework⁴ - a set of PHP classes implementing common reusable tools. A running Shibboleth service provider should be present and connected to an appropriate federation.

The login service operation proceeds as follows (see Figure 1): When an unauthenticated user wants to enter the application (1), instead of submitting a login form, the user is redirected to the AC Login Service (2). The URL of the login service is protected by Shibboleth SP, so when the user is redirected there a standard Shibboleth session creation is initiated (3) – the user is redirected to the WAYF/DS service and further to his home identity provider for authentication (4).

After successful authentication, the AC Login Service obtains the user's identity (5). Through a remote API call the login service checks, whether the user exists in the Adobe Connect database (6). If it does not (7), a new user will be created (8, 9) based on the personal data obtained from the user's attributes (givenName, sn, email).

An important part is the password generation. The user will never use the password, because as it is expected, he will never authenticate to Adobe Connect directly. So we need to set the user a password, which could not be guessed easily and at the same time the login service should be able to determine what password to use without the need of checking some other database or structure. That means, the password should be generated by a common algorithm using a piece of data as input, which is constant for each user. In our case, the algorithm is MD5 and the

³ https://vidcon.cesnet.cz/videokonference/doc/howto/adobe_connect_en/index

⁴ <http://framework.zend.com/>

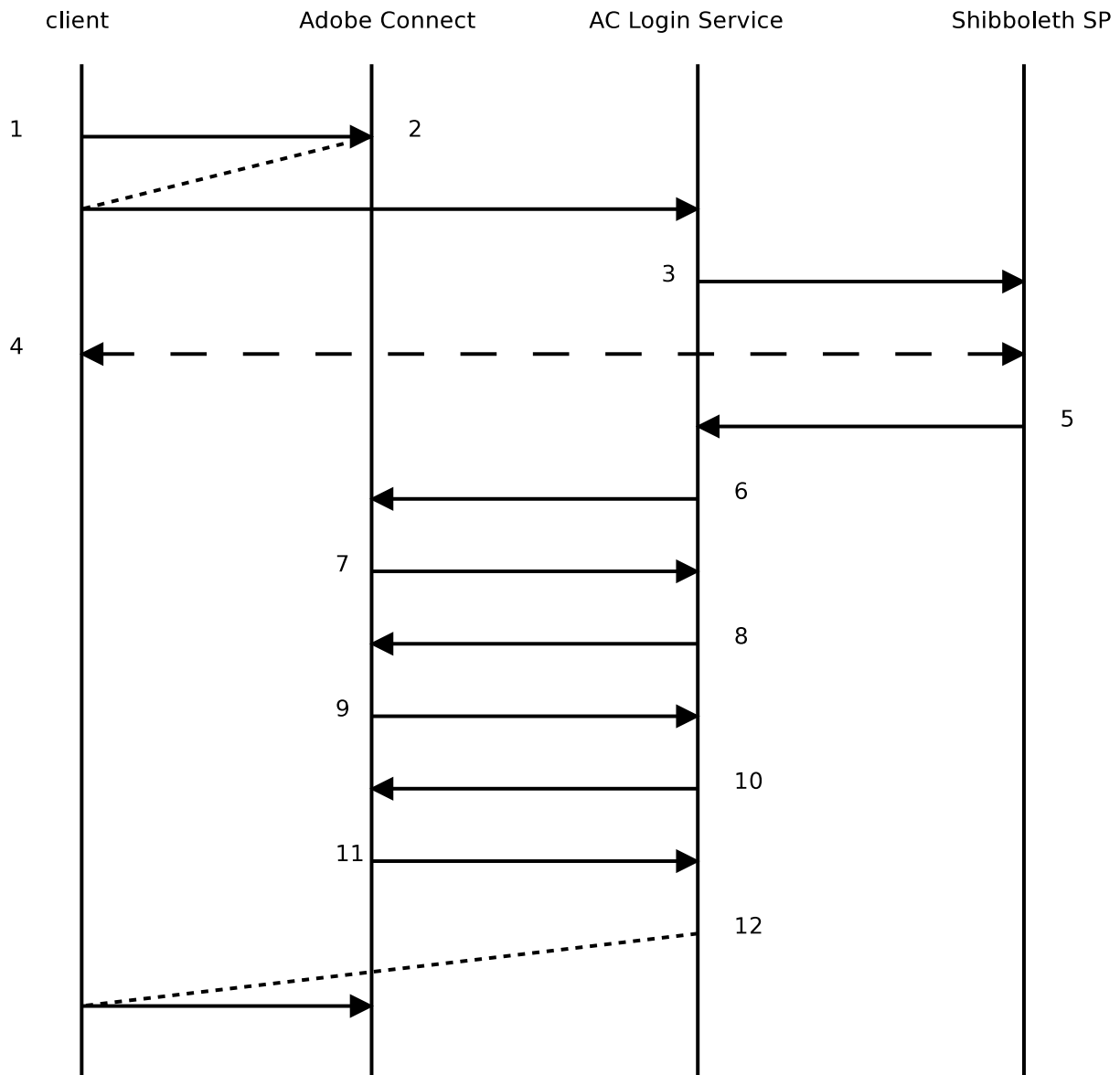


Figure 1. Communication diagram

input is the user ID (username, email). To increase security a secret salt is being used. The salt is a constant string set in the configuration. It is essential to keep that string secure, otherwise the users' passwords may be compromised.

After checking if the user exists, the login service tries to log the user in, using his user ID (username, email) and password, generated as explained in the previous paragraph (10). If the user has been logged in successfully, the server returns a response containing a session cookie (11). Otherwise an error message is returned and the login service shows it to the user. If there is no error, the whole authentication process is completely transparent for the user, who is finally redirected to the Adobe Connect server with the cookie as a GET parameter in the URL (12).

Most of the remote API calls can be performed only after authentication. Some of the remote API calls, especially the one that creates a new user, require administrator privileges. So the login service needs an administrator account to be able to function properly.

4 Installation

We need to install Zend Framework and AC PHP API library. Since they are written in PHP, in most cases it is enough to download them and unpack them in a suitable directory on the file system, for example - */var/lib/php*.

```
# cd /var/lib/php
# wget https://vidcon.cesnet.cz/_media/videokonference/doc/howto/
    adobe_connect/acapi-0.1.0.tgz
# tar xvfz acapi-0.1.0.tgz # ln -s acapi-0.1.0 acapi
```

It is convenient to place Zend Framework in a separate directory:

```
# cd /var/lib/php
# mkdir zend
# cd zend
# wget http://framework.zend.com/releases/ZendFramework-1.9.6/
    ZendFramework-1.9.6.tar.gz
# tar xvfz ZendFramework-1.9.6.tar.gz
# ln -s ZendFramework-1.9.6/library/Zend
```

The AC Login Service is being installed in a similar way. We need to download it and place it at a suitable location, for example */var/www/apps*.

```
# cd /var/www/apps
# wget https://vidcon.cesnet.cz/_media/videokonference/doc/howto/
    adobe_connect/aclogin-0.2.0.tgz
# tar xvfz aclogin-0.2.0.tgz
# ln -s aclogin-0.2.0 aclogin
```

The service is now installed in the */var/www/apps/aclogin* directory. That directory will be referred as *ACLOGIN_HOME*.

5 Apache Configuration

In Apache configuration we need to create an alias to the *ACLOGIN_HOME/www* directory and protect it with Shibboleth:

```
Alias /aclogin ACLOGIN_HOME/www
<Directory ACLOGIN_HOME/www>
    AuthType shibboleth ShibRequireSession On
    require valid-user
</Directory>
```

6 Shibboleth Configuration

The Shibboleth configuration may vary in different cases, but generally we have to set the right request and attribute mapping.

In *shibboleth2.xml* we need to set the mapping to the */aclogin* location in the *RequestMapper* element:

```

<RequestMapper type="Native">
  <RequestMap applicationId="default">
    <Host name="ac.example.org" authType="shibboleth"
      requireSession="false">
      <Path name="aclogin" requireSession="true" />
    </Host>
  </RequestMap>
</RequestMapper>

```

The following attributes must be mapped properly in the *attribute-map.xml* configuration file:

- **UID** (username or any user identifier) – generally, it is one of these attributes
 - *eduPersonPrincipalName*, *eduPersonTargetedId*
- **email**
- **first name** (givenName)
- **surname** (sn)

It does not matter what variable names those attributes are assigned. The names can be entered in the AC Login Service configuration. Just a small note about the UID attribute - in the *shibboleth2.xml* configuration file there is a specification about how to populate the REMOTE_USER environmental variable:

```

<ApplicationDefaults id="default" policyId="default"
  entityID="https://sp.example.org/shibboleth"
  homeURL="https://sp.example.org/index.html"
  REMOTE_USER="eppn persistent-id targeted-id"
  signing="false" encryption="false">

```

In this particular example (default settings) if the eppn (*eduPersonPrincipalName*) attribute is set, the REMOTE_USER variable will take its value, else it will take the *persistent-id* attribute value and so on. In the configuration of the AC Login Service there will be two choices, how to extract the user ID - directly from an attribute or the aggregated value of the REMOTE_USER variable.

7 Adobe Connect Configuration

No special configuration is needed for the Adobe Connect server. The AC Login Service requires an account with administrator privileges to be able to check users and create new ones on-the-fly.

By default, Adobe Connect uses users' emails as usernames. That should be forbidden in *Administration - Users and Groups - Edit Login and Password Policies*.

8 AC Login Service Configuration

First, the right paths to the external libraries have to be configured in the *ACLOGIN_HOME/init.php* file :

```
// The Zend framework directory
define('ZEND_FW_DIR', '/var/lib/php/zend/');
// The directory, where the AC PHP API is installed
define('ACAPI_LIB_DIR', '/var/lib/php/acapi/');
```

The configuration is stored in *ACLOGIN_HOME/config/aclogin.ini*. The configuration file doesn't exist after unpacking the application, but there is a sample *aclogin.ini.dist* file with the typical configuration with all the directives explained. It's a common INI file composed of sections with „*key = value*“ pairs. So we need to rename the file to *aclogin.ini* and edit it to suit our environment.

We need to set properly at least the following essential directives:

- section **[general]**
 - **admin_email** - the email of the person responsible for the operation of the service, the email will be shown to the user in case of an error
 - **entity_id** - the entityId of the Shibboleth SP
- section **[account]**
 - **password_salt** - a random string used in password generation
 - **redirect_uri** - the URL to redirect the user to after successful authentication, the URL of the Adobe Connect server
- section **[shibboleth]**
 - **uid_field** OR **remote_user_field** - if we want to extract the user ID from a specific attribute, we have to supply a value for the `uid_field` directive, otherwise we may use the `REMOTE_USER` environment variable and leave the default setting `remote_user_field = REMOTE_USER`
 - **mail_field** - the environment variable holding the email attribute
 - **givenName_field** - the environment variable holding the given name attribute
 - **sn_field** - the environment variable holding the surname attribute
- section **[acapi]**
 - **uri** - the complete URL, the remote Adobe Connect API is listening at
 - **username** - the username of the administrator account used for user provisioning
 - **password** - the password of the administrator account

If we want to use the logging options, we have to make sure, that the web server is allowed to write into the log file assigned in the configuration.

9 Testing the Installation

Both Apache and Shibboleth SP have to be restarted after modifying their configuration files. To test the installation, we have to visit *https://HOSTNAME/aclogin/* (*HOSTNAME* is the domain name or address of the server, the login application is installed on). If the application works properly, a standard Shibboleth session should be initiated (redirection to the WAYF/DS service and further to the home IdP). Upon successful login we should end logged in Adobe Connect Pro.

10 Running in Production Mode

Before turning the AC Login Service into production mode it is necessary to modify the Adobe Connect login page. We do not want users to log in directly any more. Actually, they will not be able to do it. That is why the login form should be removed from the page. Instead, we have to place a login button, which redirects the user to the AC Login Service. If it is still necessary anonymous users to be able to use Adobe Connect, we should leave the option for guest access. Of course, it is possible to combine Shibboleth access with local access, but both options should be displayed clearly to the user.

The Adobe Connect user interface is defined in different XSL files. The login form definition is placed in *BREEZE_HOME\appserv\apps\system\login.xsl*. After modifying the contents of any XSL file we have to restart Adobe Connect server, in order to activate the changes. The login button may be accompanied by the logo of the federation (Figure 2).

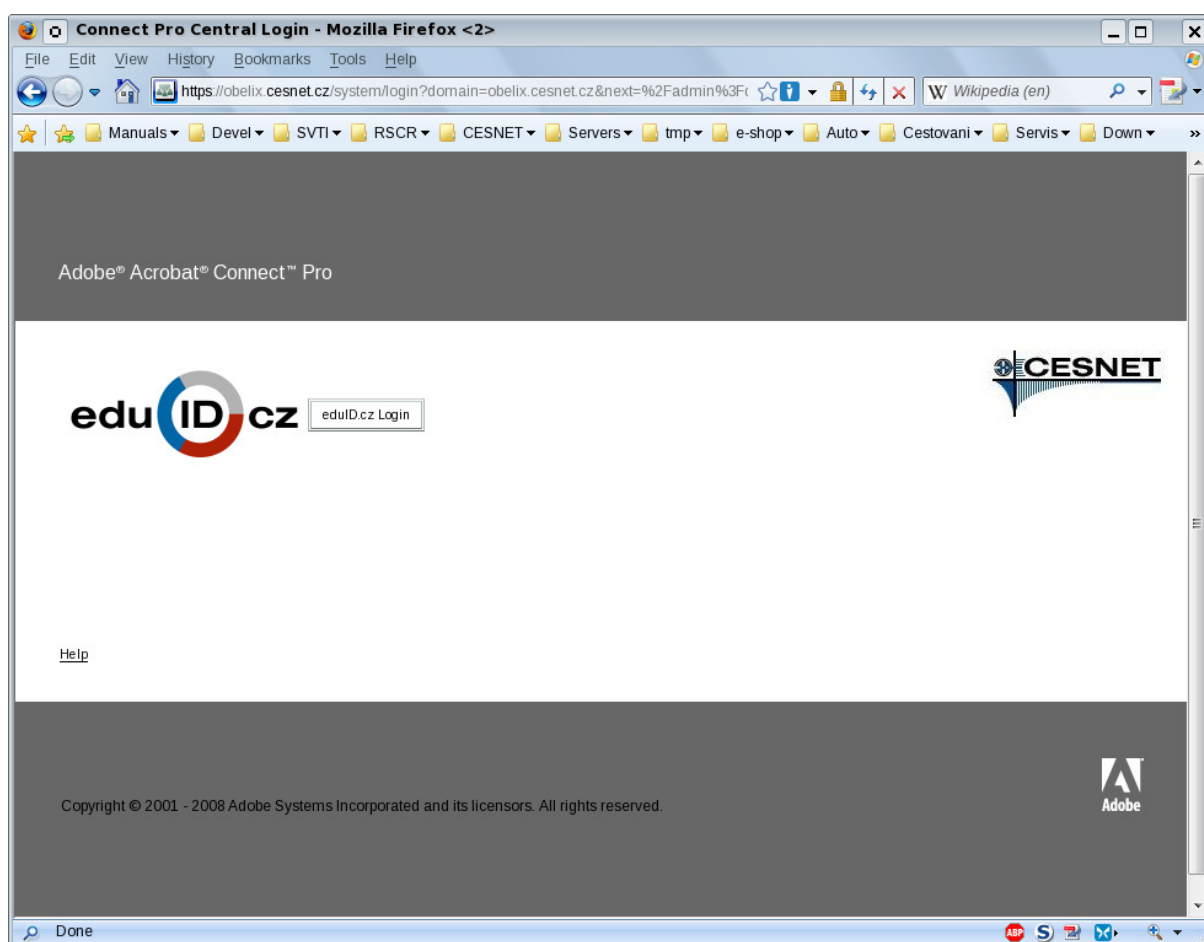


Figure 2. Adobe Connect Pro login page

We can also customize the error page, which appears in case of problems during the authentication. To do this, we need to edit the layout template file – *ACLOGIN_HOME/tpl/default/layouts/layout.phtml*.

11 Further Improvements

The AC Login Service is fairly simple application, which can be further improved and extended easily. For example, more user provisioning mechanisms may be implemented. Users may be assigned different groups and may be granted extended privileges based on their attributes. The additional remote API calls can be easily implemented using the generic AC PHP API library. The library is not coupled with the login service and may be used separately in other applications.

References

- [1] *Apache HTTP Server Version 2.2 Documentation*. Apache Software Foundation, 2009 [cit. 2009-12-16]. Available online⁵.
- [2] *Using Adobe Acrobat Connect Pro 7.5 Web Services*. Adobe Systems, Inc. [cit. 2009-12-16]. Available online⁶.

⁵ <http://httpd.apache.org/docs/2.2/>

⁶ http://help.adobe.com/en_US/AcrobatConnectPro/7.5/WebServices/index.html