Shibboleth based security for Xtremweb-CH: internal report

Marko Niinimaki, Mohamed Ben Belgacem
HEPIA
September 2009, updated Nov 2010.

Introduction

As the classic definition goes, data security is the science of protecting data in computer and communication systems from unauthorized disclosure and modification [Den1982]. Authentication (identity verification), authorization and accounting [RFC2904] are often seen as the cornerstones of a security architecture.

In the Xtremweb-CH (XWCH) software [AB2006] (see http://www.xtremwebch.net), the focus has been in creating an efficient platform of peer-to-peer distributed computing. Two separate issues are of interest here, (1) security when using the XWCH web GUI (2) job submission from a client to Xtremweb-CH coordinator.

In the previous versions of Xtremweb-CH, all the authentication and authorization were local username/password based. In order to provide the platform for wider audience and to minimize local administration overhead, a multi-organization/multi-service single-sign-on system (Shibboleth) is introduced here. Shibboleth operates over the https that uses a secure layer over http [RFC2818], encrypting the content of the messages. In practice this means that an intruder cannot read the contets of the communication using a simple network data logger.

The rest of this report is organised as follows: Section 1 presents the main components of XWCH, Section 2 presents Shibboleth and Section 3 its implementation for logging into the XWCH coordinator. Section 4 discusses using HTTPS instead of HTTP in API calls.

1. The main components and user interaction of Xtremweb-CH

Xtremweb-CH consists of a coordinator, warehouses and workers. Warehouses are used by workers to share information and store input and output files. The coordinator accepts jobs from clients, schedules them to workers, and forwards their results to the users. Currently, the users submit jobs by an API, but a GUI based job submission in being considered. This is made possible by the fact that the coordinator (running as a web service) is the only component with which the user communicates. Having a single single-sign-on mechanism on the coordinator will be beneficial to the user in two senses

- no need to remember additional user names/passwords.
- no need to install local software in order to manage jobs by GUI.

2. SWITCH AAI Shibboleth as login method

Currently, a user must register with the Xtremweb-CH administrator in order to receive a user ID. The ID should be used as a parameter in job submission. For an adminstrator, this approach is straightforward, since the only administrative operations are to allow or deny job submissions from a given ID.

Shibboleth [EC2005] is a tool for federated identity management and access control, used by many collaborations including SWITCH. With Shibboleth, a user from an collaborating instituted is allowed an access to a collaborator's web site (restrictions can apply), after he/she is authenticated at the home institute. In the Shibboleth terminology, the Xtremweb-CH web site, www.xtremwebch.net is a *service provider* that is accepted by an *identity provider* (in our case HES-SO) that is a part of the SWITCH AAI Shibboleth community. After the service provider has been accepted, it is available to the community members. However, the decision of the authorization remains within the service. In the case of Xtremweb-CH, users of the SWITCH AAI community will be able to use their Shibboleth credentials (username, password) to connect to the Xtremweb-CH coordinator, but they will not be authorized to submit jobs unless they are authorized by the local administrator.

In the Shibboleth login to Xtremweb-CH, the user is recognized by three items that are provided by the Shibboleth identity provider: Shib-Person-ID, homeorganization, and the identity provider's URL.

3. Implementation of Shibboleth for the XWCH coordinator

The following additions have been introduced in the system.

GlassFish configuration and Apache Shibboleth integration

The Xtremweb-CH coordinator is a Java Enterprise application running under GlassFish server. The Shibboleth service is running under the Apache http server, and it has been registered at https://rr.aai.switch.ch. In order to bridge the gap between Apache and the coordinator program, the following setup is used.

- When the user arrives at a Shibboleth-protected page like https://www.xtremwebch.net/shibboleth, we can be sure that his/her credentials have been verified by his/her identity provider. The user's identity is represented to Apache as enviroment variables. Of these, we select the user ID, home organization, and the identity provider's URL. They are stored as HTTP cookies and a HTTP refresh forwards the user's browser to the Xtremweb-CH coordinator application login page.
- The code on the login page verifies that the referer (page where the user arrived) is under https://www.xtremwebch.net/shibboleth. This is needed to ensure that the cookies were not custom made in another server in order to fake the user's identity. After that, the login page tries to log in the user with his/her user ID/home organization combination. If this is successfull, the user is redirected to the home page (see Fig 3 below).

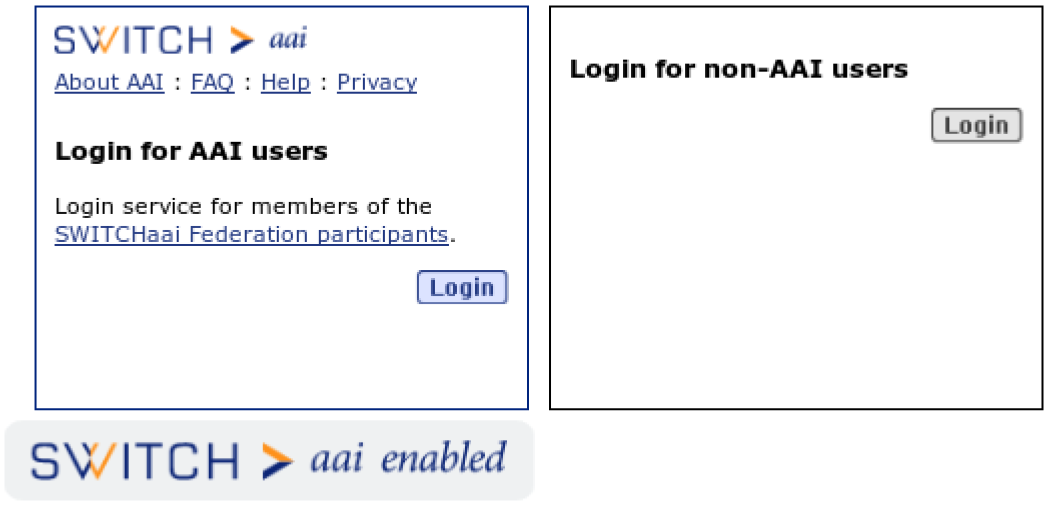
Enabling the home institute's (HES-SO) Shibboleth identity provider for www.xtremwebch.net/shibboleth

This was done using the registration form at https://rr.aai.switch.ch, and with help of the HES-SO system administrators.

A web based job submission form

A web based form currently enables the user to upload his/her application as a zip file.

Screenshots of the login process

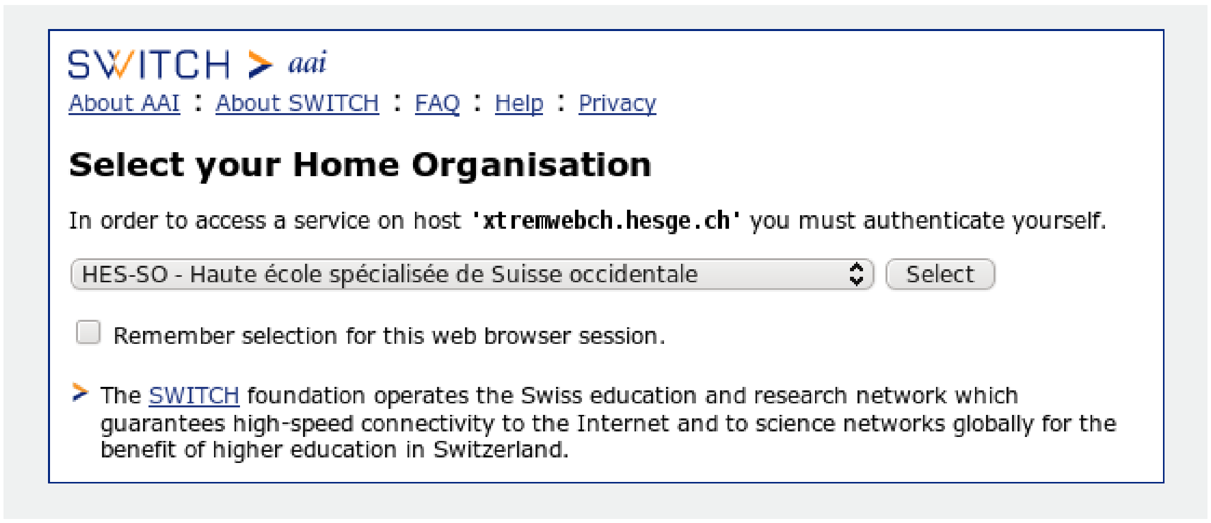

SWITCH > aai
About AAI : FAQ : Help : Privacy

Login for AAI users

Login service for members of the SWITCHaai Federation participants.

Login for non-AAI users

The login screen. Two methods of login are still possible, since the local administrator needs a non-AAI login.


SWITCH > aai
About AAI : About SWITCH : FAQ : Help : Privacy

Select your Home Organisation

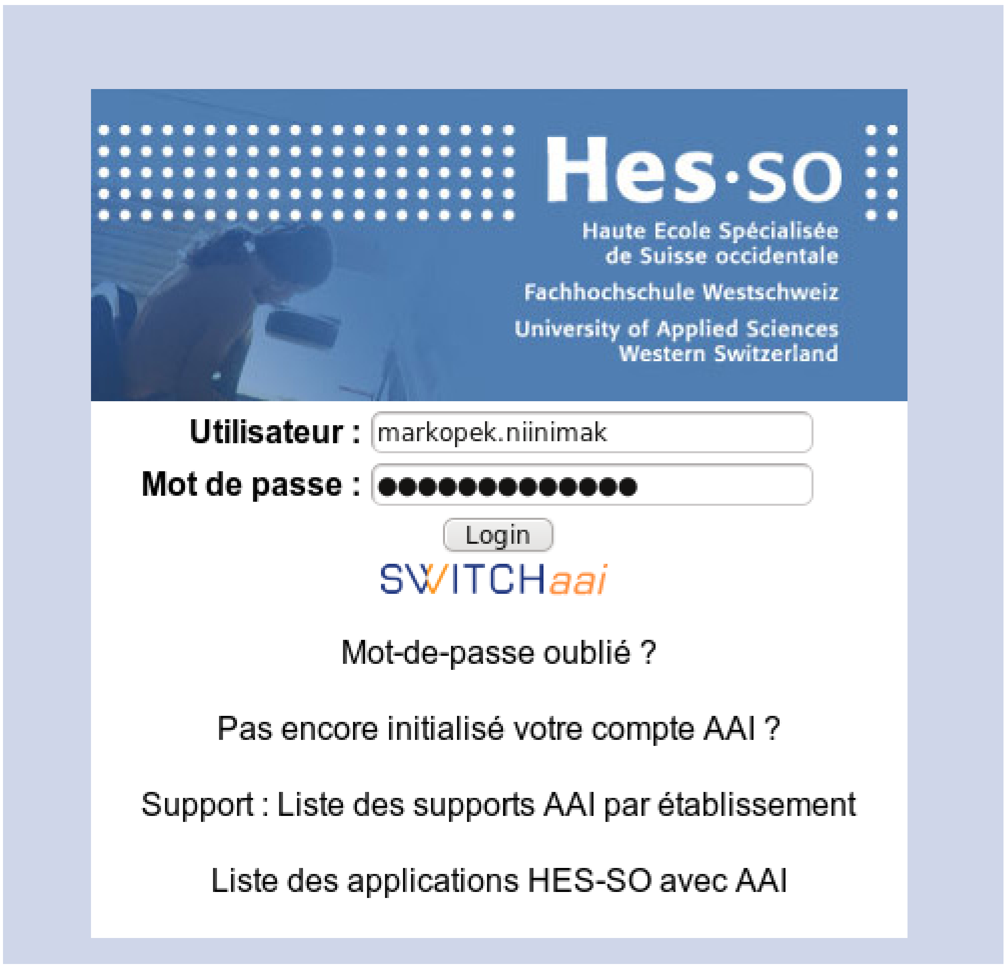
In order to access a service on host 'xtremwebch.hesge.ch' you must authenticate yourself.

HES-SO - Haute école spécialisée de Suisse occidentale

☐ Remember selection for this web browser session.

The SWITCH foundation operates the Swiss education and research network which guarantees high-speed connectivity to the Internet and to science networks globally for the benefit of higher education in Switzerland.

1. From "Login for AAI users", a redirection to identity provider (home organisation) selection.


Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences Western Switzerland

Utilisateur : markopek.niinimak

Mot de passe : ●●●●●●●●●●●●●

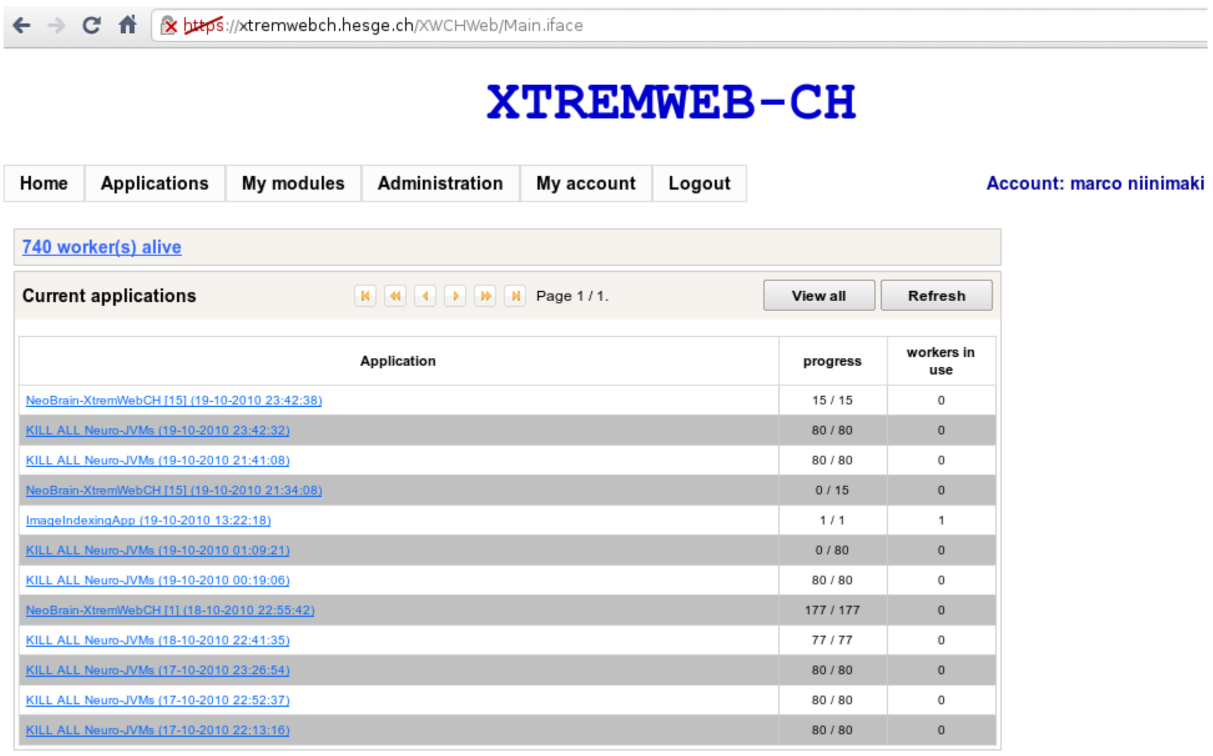
Mot-de-passe oublié ?

Pas encore initialisé votre compte AAI ?

Support : Liste des supports AAI par établissement

Liste des applications HES-SO avec AAI

2. Home organization login



Application	progress	workers in use
NeoBrain-XtremWebCH [15] (19-10-2010 23:42:38)	15 / 15	0
KILL ALL Neuro-JVMs (19-10-2010 23:42:32)	80 / 80	0
KILL ALL Neuro-JVMs (19-10-2010 21:41:08)	80 / 80	0
NeoBrain-XtremWebCH [15] (19-10-2010 21:34:08)	0 / 15	0
ImageIndexingApp (19-10-2010 13:22:18)	1 / 1	1
KILL ALL Neuro-JVMs (19-10-2010 01:09:21)	0 / 80	0
KILL ALL Neuro-JVMs (19-10-2010 00:19:06)	80 / 80	0
NeoBrain-XtremWebCH [1] (18-10-2010 22:55:42)	177 / 177	0
KILL ALL Neuro-JVMs (18-10-2010 22:41:35)	77 / 77	0
KILL ALL Neuro-JVMs (17-10-2010 23:26:54)	80 / 80	0
KILL ALL Neuro-JVMs (17-10-2010 22:52:37)	80 / 80	0
KILL ALL Neuro-JVMs (17-10-2010 22:13:16)	80 / 80	0

3. Automatic redirection: Logged in to XWCHWeb with Shibboleth credentials.

4. HTTPS as communication method in client - coordinator communication

In client - coordinator communication XWCH uses web service calls that run typically over HTTP. However, in the previous section we saw that we provide the XWCH Coordinator's Web GUI over HTTPS. Therefore, we can provide the web services using the same method.

Using the XWCH Java API, a simple communication with the coordinator is started as in Figure 4.

```
ServerAddress = "http://xtremwebch.hesge.ch";
IdClient = "xxxxx";
XWCHClient c = new XWCHClient(ServerAddress, ".", IdClient);
if (!c.Init()) { System.out.println("Init error");  throw new Exception();  }
if (!c.PingWarehouse()) {  System.out.println("Warehouse error");  throw new
Exception(); }
String appid = c.AddApplication("HelloXWCH");
```

4. XWCHDemo program that uses XWCH Java API

There, IdClient is the user's ID and c.Init(), c.PingWarehouse() and c.PingWarehouse() are eventually implemented as web service calls.

Let us assume that the code above forms the core of a Java program called XWCHDemo. In order to use "https://xtremwebch.hesge.ch" instead of http://xtremwebch.hesge.ch" as a server address we need the client side to trust the server's certificate. This is done by the following trivial method:

- The server's public certificate is stored in a Java certificate store (a freely available program called InstallCert can be used for this)
- When XWCHDemo is invoked, the certificate store is passed as a special parameter: java -Djavax.net.ssl.trustStore=jssecacerts -classpath .:XWCHClientAPI.jar XWCHDemo

When using HTTPS, the communication between the client and server is encrypted and each message contains a certificate identifying the correspondent. This is naturally more time-consuming than the plain-text HTTP communication. In order to test how much overhead the HTTPS communication adds, we have tested the code in Figure 4 in the following environment:

- A coordinator, a warehouse and a worker in a modern i386 desktop computer (2 dual-core Intel processors, 3 GB memory)
- Linux Ubuntu, Sun Java 1.6, Apache 2, Shibboleth 2.1, GlassFish 2, XWCH version 20-10-2010 (worker version 1.08).
- Running 100 times (sequentially) a program that creates an application, uses a module and sends an empty input file. NB: we do not wait that the task that is generated by the module is started or finished -- this is a matter of the coordinator - worker -communication and not our concern in this report.

The results are as follows:
1) With HTTP 1m36.199s 2) With HTTPS 5m2.204s

Appendix: Implementation notes

This section discusses the implementation steps taken to enable Shibboleth authentication for www.xtremwebch.net in the specific environment of HEPIA, SWITCH AAI, and a Fedora Core 5 Linux server (64bit).

Apache as gatekeeper

The basic idea is to use the Shibboleth variables "Shib-Identity-Provider", "Shib-Person-uid" and "homeOrganization" in the application from http cookies. The following code in /var/www/shibboleth/index.php gets them from the Apche server and inserts them in the browser as cookies.

```
<?
$idprov = $_SERVER['Shib-Identity-Provider'];
$personuid = $_SERVER['Shib-Person-uid'];
$homeorg = $_SERVER['homeOrganization'];
$dummy = setcookie('Shib-Identity-Provider', $idprov, 0, '/');
$dummy = setcookie('Shib-Person-uid', $personuid, 0, '/');
$dummy = setcookie('homeOrganization', $homeorg, 0, '/');
?>
```

Changes in XWCHWeb source code

Now that the Shibboleth data is in the browser cookies, in can be read by XWCHWeb, as follows (XWCHWeb_Entreprise/src/java/xwchweb/Login.java):

```
String mypuid = null; String myorg = null; String myprov = null;
FacesContext fci = javax.faces.context.FacesContext.getCurrentInstance();
```

```
ExternalContext ec = fci.getExternalContext();
Map cookiemap = ec.getRequestCookieMap(); Iterator it =
cookiemap.entrySet().iterator();
while (it.hasNext()) {
    Map.Entry pair = (Map.Entry)it.next();
    String mykey = pair.getKey().toString();
    Cookie mycookie = (Cookie)pair.getValue();
    String myval = mycookie.getValue();
    if (mykey.equals("Shib-Person-uid")) mypuid = myval;
    if (mykey.equals("homeOrganization")) myorg = myval;
    if (mykey.equals("Shib-Identity-Provider")) myprov = myval;
}
```

For the time being, the XWCH identity of the Shibboleth user is simply "Shib-Person-uid"@"homeOrganization". The CLIENT database table contains user information in the following form:
client-id, non, prenom, email, isvalidated, isadmin.

The field checked in the Shibboleth based login is *email* (in XWCHWeb_Entreprise/src/java/xwchweb/Login.java, function btnShiblogin_action). If it matches with "Shib-Person-uid"@"homeOrganization", the login is accepted and the user forwarded to the home page.

Please note that:

- the admin needs to create users with matching email for the shibboleth login to work
- in EJB/src/java/webapplication1/ext/bean/clientFacadeExt.java, a demo user is added in function CheckAccount

Shibboleth installation at www.xtremwebch.net

Adopted from https://www.switch.ch/aai/docs/shibboleth/SWITCH/2.1/sp/deployment/debian-etch-source.html

Shibboleth software compilation and installation as in debian-etch-source.html, with following exceptions:
/etc/init.d/shibd copied from shibboleth-2.1/configs/shibd-redhat, needs the following changes:

```
#beginning of file
if [ -d /var/run/shibboleth ]; then echo exists > /dev/null; else mkdir /var/run/shibboleth; fi
#daemon start
config=/etc/shibboleth2/shibboleth2.xml
su - $SHIBD_USER -c "$shibd -p $pidfile -c $config -f &"
```

Apache configuration:
Copying the compiled module to Apache's directory: /usr/lib64/httpd/modules/mod_shib_22.so
The following changes in /etc/httpd/conf.d/ssl.conf:

```
#beginning of file
LoadModule mod_shib modules/mod_shib_22.so
# inside the VirtualHost tag, important
ShibConfig /etc/shibboleth2/shibboleth2.xml

<Location /shibboleth-sp>
 Allow from all
```

```
</Location>

Alias /shibboleth-sp/main.css /opt/shibboleth-sp2/share/doc/shibboleth/main.css
Alias /shibboleth-sp/logo.jpg /opt/shibboleth-sp2/share/doc/shibboleth/logo.jpg

<Location /shibboleth>
 Options Indexes FollowSymLinks MultiViews ExecCGI
 AllowOverride All
 Order allow,deny
 allow from all
 AuthType shibboleth
 ShibRequireSession On
 ShibRedirectToSSL 443
 require valid-user
</Location>
```

Service registration with SWITCH

As in the instructions of https://www.switch.ch/aai/docs/shibboleth/SWITCH/2.1/sp/deployment/debian-etch-source.html and the actual registration at https://rr.aai.switch.ch

References

[AB2006] N. Abdennadher, R. Boesch: A Scheduling Algorithm for High Performance Peer-To-Peer Platform, CoreGrid 2006.
[Den1982] D. Denning: Cryptography and Data Security, Addison-Wesley, 1982.
[EC2005] M. Erdos, S. Cantor: The Shibboleth Architecture, http://shibboleth.internet2.edu/, 2005.
[RFC2904] J. Vollbrecht et al: RFC2904 - AAA Authorization Framework, 2000.
[RFC2818] The Internet Society: HTTP over TLS, 2000.