Installing a Shibboleth Target

Table of Contents

Introduction

This document aims to set out what an institution needs to do from scratch to install a Shibboleth Target . Generally, this requires the use of attributes for users.

Authors

This document was compiled by Caroline Ingram, with contributions from Malcolm Murray and Michael Young, at Durham University, and Cal Racey, Jon Dowland and Janet Wheeler, at Newcastle University.

Purpose

The purpose of this document is to lower the skill set required to understand what is needed to install a Shibboleth Target.

This document follows from previous documentation available from http://iamsect.ncl.ac.uk/.

About Shibboleth

Shibboleth is a single sign on project that has been developed by a federation of higher education establishments in the United States called Internet2. Shibboleth is not an authentication or authorisation

scheme. It is an open, standards-based protocol for securely transferring attributes between an identity provider (local institution) site and service provider (resources) site which is supplied as an open-source reference software implementation. The current reference installation is Shibboleth 1.2. This document will be reviewed for Shibboleth 1.3 during 2006.

About Access Control

Academic libraries licence and offer access to a range of online resources that are regarded as necessary to support research and teaching in any subject. One of the big issues to be solved is maintaining a balance between adhering to the legal and contractual responsibilities to publishers (to limit access to only those users covered by licence terms) and to the library's users (to keep personal information secure), and, finally, to ensure the fundamental function of a library operates - that users are shown the simplest path to the information they want[1].

Skills Set required

In order to Install and manage a Shibboleth software implementation you will need to be able to access the following skills:

- A reasonable working knowledge of the Linux (or Unix) Command Line Interface (CLI);
- Knowledge of how to use the apache web server, either the 2.0.* or 1.3.* versions;
- Familiarity with the concepts of https communication (certificates, keys and the like);
- Familiarity with firewalls or access to someone who is familiar, in particular with Linux iptables style firewalls;
- Familiarity with the setup of Windows Active directory, or access to someone who has those skills.

Finally, you will need a willingness to read around subject areas, management pages and mailing lists, since this is an emerging and swiftly developing area.

Planning

There are several essential elements that must be present in the environment to ensure Shibboleth functions well (see the section called “Installing a Shibboleth Target”). Shibboleth is entirely written in Java on the identity provider side. The basic installation of the Shibboleth Identity Provider Server should be carried out before joining the federation, as the latter will test that the installation works, and will require information that will not be available until installation is complete.

IAMSECT deliverables include a Shibboleth Identity Provider set-up guide.

Install Pre-requisites

Check previous installation documents on the IAMSECT website for more details. The following need to be considered:

- Unless otherwise stated, do not build and install the components in this guide under a remote branch

[1]For additional information see Paschoud, J. (2005) Shibboleth and SAML: at last, a viable global standard for resource access management. *New Review of Information Networking* Vol. 10, No. 2. (November 2004), pp. 147-160

of the file system;

- configure your environment so that tools such as wget are able to access the web:

  ```
  $ export http_proxy=webproxy.example.edu:8080
  ```

  Where `webproxy.example.edu` is the host and `8080` the port of your web proxy;

- have a firewall installed on servers, but note the ports that need to be unblocked;
- note that Pubcookie requires accurate network time.

Installing a Shibboleth Target

Installing a Shibboleth Target can be broken down into three stages. The installation results in getting a service provider to use Shibboleth in order to identify users and authorise use.

Installing the software

Installing Shibboleth software has been convered in more detail in other documentation produced by the IAMSECT project[2]. The following steps will allow you to complete an installation from source.

Sources and software

The SWITCH install documentation[3] offers a succinct set of instructions for installing the pre-requisite software. There is little there which is debian-specific. The following software can be installed from the vendor or the software author:

- gcc, g++, make etc. – to build the source packages
- apache(-dev) : i.e. webserver + apxs (either 1 or 2)
- libssl(-dev)
- libcurl
- libxml-security-c
- mod_ssl

The following software must be obtained from http://wayf.internet2.edu/shibboleth/

- log4cpp – via shib because the project is in a ‘limbo state’
- xerces-c – latest official xerces-c release has a bug which breaks shib. This version doesn’t
- opensaml
- shibboleth

[2] IAMSECT project documentation is available from http://iamsect.ncl.ac.uk/deliverables/
[3] http://www.switch.ch/aai/docs/shibboleth/SWITCH/1.2/install-target-1.2.1-debian.html

Installation order

The software must be installed in the following order: first gcc, g++, make etc., then apache + apache-dev (axps), followed by libssl + libssl-dev (headers), and finally libcurl.

Debian:

```
# aptitude install c++-compiler make apache apache-dev libssl-dev
libcurl-dev libapache-mod-ssl
```

Redhat/fedora:

```
# up2date g++ gcc make httpd httpd-devel openssl openssl-devel
curl curl-devel mod_ssl
```

Next, prepare the directory where the other software goes

```
# mkdir -p /opt/shibboleth-1.2.1
# cd /opt
# ln -s shibboleth-1.2.1 shibboleth
# cd shibboleth
```

log4cpp

```
# wget http://wayf.internet2.edu/shibboleth/log4cpp-0.3.5rc1.tar.gz
# tar xvzf log4cpp-0.3.5rc1.tar.gz # cd log4cpp-0.3.5rc1
# ./configure --prefix=/opt/shibboleth --with-pthreads --disable-static --disable-
# make && make install
```

xerces-c

```
# wget http://wayf.internet2.edu/shibboleth/xerces-c-src_2_6_1.tar.gz
# tar xvzf xerces-c-src_2_6_1.tar.gz
# cd xerces-c-src_2_6_1
# export XERCESCROOT=`pwd`
# cd src/xercesc
# ./runConfigure -p linux -r pthread -P /opt/shibboleth
# make && make install
# cd ../../..
```

Note

The -c and -x args, defining the C and C++ compilers, can be dropped if they are the default ones on the system anyway (and should be on any of the targets to set up)

libxml-security-c

```
# wget \
http://xml.apache.org/dist/security/c-library/old/xml-security-c-1.1.0.tar.gz
# tar xvzf xml-security-c-1.1.0.tar.gz
# cd xml-security-c-1.1.0/src
# ./configure --prefix=/opt/shibboleth --without-xalan
# make && make install
# cd ../..
```

Note

The above relies on the variable $XERCESCROOT defined in the previous section, just in case you do things in a different order
opensaml

```
# wget http://wayf.internet2.edu/shibboleth/archive/opensaml-1.0.1.tar.gz
# tar xvzf opensaml-1.0.1.tar.gz # cd opensaml-1.0.1
# ./configure --prefix=/opt/shibboleth   \
             --with-curl=/opt/shibboleth \
             --with-log4cpp=/opt/shibboleth
# make && make install
# cd ..
```

shibboleth

```
# wget http://wayf.internet2.edu/shibboleth/archive/shibboleth-1.2.1a.tar.gz
# tar xvzf shibboleth-1.2.1a.tar.gz
# cd shibboleth-1.2.1
# ./configure --prefix=/opt/shibboleth \
              --enable-apache-20 \
              --with-apxs2 \
              --with-log4cpp=/opt/shibboleth-1.2.1
# make && make install
```

Note

for apache 1.x, use `--enable-apache --with-apxs` rather than `--enable-apache-20 --with-apxs2`.

Test install

Visit the URL of your protected resource. You should be sent off to your origin and asked to log in. On successful login, you should be directed back towards the target, and see the desired page. If the desired page prints out environment variables as suggested, you should see some that correspond to attributes that the origin is releasing.

Setting up a dummy service

In order to test the target software, you need a site setup to protect. A simple ‘hello world’ page would

suffice but isn't very interesting. You could use the following to provide a shibboleth-protected server status page:

```
<Location /server-status>
 SetHandler server-status
 AuthType shibboleth
 ShibRequireSession On
 require valid-user
</Location>
```

Another good choice would be a CGI or PHP script (or similar) which will print out the environment variables. The following PHP script for example

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

Then, the attributes that are provided to the application will be visible on the page.

After following this configuration list, test install again, as in the section called "Test install".

Getting the Shibboleth configuration right

ADD TEXT: what are we configuring? what are our expectations for the target?

Log permissions

```
root# cd /opt/shibboleth-1.2.1/var/log
root# chown root:apache shibboleth
root# chmod 775 shibboleth
```

Where 'apache' is the username the httpd runs under. `apache' on redhat; `www-data' on debian; /opt/shibboleth-1.2.1 is the root under which the shibboleth-specific software was installed.

Paths

Type the following in .profile:

```
source ~/.bashrc
```

Stick the following in .bashrc:

```
export PATH=/opt/shibboleth/bin:$PATH
```

Initialisation scripts

The system init scripts need adjusting, so that:

- apache is able to load the Shibboleth libraries on startup
- apache is started after the shibd component of Shibboleth

This is achieved by modifying the `/etc/init.d/apache`script (plus indicates added lines):

```
...
start() }
 echo -n $"Starting $prog: "
 check13 || exit 1
+ export LD_LIBRARY_PATH=/opt/shibboleth/lib
 LANG=$HTTPD_LANG daemon $httpd $OPTIONS
 RETVAL=$?
 echo
...
```

shibd init script

An example init script for starting and stopping the shibboleth daemon component can be found in the Appendix. see the section called “Sample shibd init script”.

More apache

Include configuration items relevant to shibboleth

```
ln -s /opt/shibboleth/etc/shibboleth/apache2.config \
/etc/httpd/conf.d/shibboleth.conf
```

If you find apache won’t start up and you get `permission denied’ errors, check the section called “Error Messages and Solutions”.

Keys

There should be a `/etc/httpd/conf.d/ssl.conf` file. This defines SSL settings including certificate and private key. Visiting `https://yourserver/` should bring up the redhat test page, after you’ve clicked past some certificate warnings.

You are going to want to change these keys for something else, either real keys provided by a top-level Certificate Authority, keys provided by a federation you are operating within (e.g. SDSS Test Federation) or a self-signed or internally-issued set.

Once you’ve obtained your replacement key and certificate, change the `ssl.conf` file to reference them and restart apache. Re-visit the https url and look at the cert in the browser. If necessary, once you’ve confirmed it's the right one, add it to your browser’s list of trusted certificates permanently. This will reduce the number of pop-up windows that occur during testing.

Shibboleth

- WAYF URL

As suggested by the `shibboleth.xml` file, the default wayfURL is the *In Queue* federation's service. Change to https://your-handle-server/shibboleth/HS for internal testing against your own origin.

```
-->
                 <Sessions lifetime="7200" timeout="3600"
checkAddress="true"

- wayfURL="https://wayf.internet2.edu/InQueue/WAYF"
+ wayfURL="https://lock.ncl.ac.uk/shibboleth/HS"

 shireURL="/Shibboleth.shire"
shireSSL="false"/>
<!--
```

- choice of providerId

  The main application element's providerId attribute must be changed to a URI chosen by the deployer to uniquely identify the service.

  i.e. in our example, providerId becomes providerId="https://webdev2.ncl.ac.uk/shibboleth/target"

  Note

  The providerId must match the the Name attribute in the `Destinationsite` tag in the federation's `sites.xml` (or the `sites.xml` that the origin you are using has)

- target certificates

  The target, by default, uses shibd.key and shibd.crt supplied in the shibboleth source. We need to get hold of a new keypair which is more secure and trusted by the origin. You may get one from a federation provider, or a CA which is trusted by your federation, or you could use Self Signed Certificates. From http://shibboleth.internet2.edu/guides/deploy-guide-target.html:

  - The only target component that must have a private key and certificate is the shibd.
  - it is mandatory for the shibd to authenticate when contacting an AA
  - certificate is signed by a CA accepted by the origin site
  - The shibd is assigned a key and a certificate using shibboleth.xml's File Resolver element

  To make the shibd use your new keypair:

```
 <Credentials
xmlns="urn:mace:shibboleth:credentials:1.0">
 <FileResolver Id="defcreds">
 <Key format="PEM">
- <Path>/opt/shibboleth-1.2.1/etc/shibboleth/shibd.key</Path>
+ <Path>/opt/shibboleth/self-signed/ssl.key</Path>
 </Key>
 <Certificate format="PEM">
- <Path>/opt/shibboleth-1.2.1/etc/shibboleth/shibd.crt</Path>
+ <Path>/opt/shibboleth/self-signed/ssl.crt</Path>
 </Certificate>
 </FileResolver>
```

We have omitted how to make the IdP trust these keys. In most cases it is not necessary as you will end up using the Federation (in this case SDSS-provided) keys. In this case also the SDSS CA was already trusted by our origin.

"Federation Provider and Trust Provider elements must be added or changed as needed to reflect the arrangements required. This information is often provided by federations to their members"[4]

Getting the service to work with Shibboleth

Getting the service (Blackboard, Zope) to accept information from shibboleth and use it in some way: i.e. identify the user by their attributes, etc., is the final step in the set up. The amount of work involved in getting a particular service to work with Shibboleth is completely dependent on the service. This article goes on to decribe (in sections 3 and 4) how this process works for two VLEs, Blackboard and Zope.

Deployment of the Blackboard VLE as a Shibboleth target

Background

As mentioned in the section called "Planning", the first task is to set up a Shibboleth Identity Provider (Origin) Server including PubCookie. This documentation assumes that the URL for your Blackboard server is: http://blackboard.uni.ac.uk/

The basic steps in the process are to set up a Shibboleth Target for the Blackboard server; join it to a federation if required; enable SSL encryption on the Blackboard server;

Note

If you opt for joining a federation, ensure that you SSL certificate provider (e.g. Globalsign) is supported by the federation
modify the Blackboard configuration files, specifying the Shibboleth Identity Provider Server; and finally change the authentication method to Shibboleth in the Blackboard system administration panel.

Demo integrating Blackboard with Shibboleth

Blackboard has been working on enabling its course management software to use Shibboleth as an authentication and attribute transport tool. There is a description on the Internet2 website which guides you through a demo with a 'shibbolised' Blackboard server[5]. As yet, Blackboard is not ready for a production Shibboleth service.

Blackboard White Paper 2004

In order to maximise portability, enterprise systems must be able to seamlessly exchange data in a secure manner. The Internet2 Middleware Working Group on Shibboleth has been working to test and re-

[4] Quote from http://sdss.ac.uk/wiki/wiki.pl?SetupServiceProvider
[5] http://shibboleth.internet2.edu/docs/demo-instructions.html

fine the ability for learners and instructors to seamlessly move from one learning platform to another across campuses and3 geographic boundaries without having to re-authenticate.

The Blackboard Learning System includes out of the box support for Enterprise interoperability - the ability to integrate Blackboard with student information systems, central authentication systems and library systems; and to shibde information about users, courses, enrolment records and grades. Blackboard's ability to support back office integration follows several standards, including the IMS Enterprise Specification, authentication integration with LDAP, Kerberos and DCE as well as Internet2 Shibboleth integration, see: http://www.blackboard.com/docs/AS/Blackboard_Whitepaper_Standards_QE.pdf

Obtaining a certificate from chosen Certificate Authority

The basic steps to obtaining a certificate are as follows:

- Log in to the Blackboard Web server as root
- Check to see if openSSL is already installed on your system. Ensure that the PATH includes the openSSL code provided by Blackboard, or a compatible version:

  ```
  PATH=/usr/local/blackboard/apps/openssl/bin:$PATH export PATH
  ```

- If it does not already exist, create the following directory for storing your new certificates:

  ```
  mkdir /usr/local/blackboard/apps/httpd/conf/certs
  ```

- Then change to this directory:

  ```
  cd /usr/local/blackboard/apps/httpd/conf/certs
  ```

- Generate an unencrypted RSA private key – back it up and keep this file safe it should only be readable by root

  ```
  openssl genrsa -out server.key 1024
  ```

- Create a certificate signing request (CSR) for the server RSA private key - use instructions from the CA or run the following command which will ask you several questions – the common name should be the machine name as it will appear in the URL of the Blackboard server e.g. 'blackboard.uni.ac.uk' in our example

  ```
  openssl req -new -key server.key -out server.csr
  ```

  Depending on the requirement of the CA you may have to set an expiry date (e.g. 365 days) at this point. GlobalSign did not require this, but if your CA does, run this command instead:

  ```
  openssl req -new –days 365 -key server.key -out server.csr
  ```

- Then read and check the details you put in the signing request using:

  ```
  openssl req -noout -text -in server.csr
  ```

- Finally, send this file server.csr to the CA for signing. They should generate a server.crt file for you. You now have your signed certificate.

Instructions on configuring Blackboard (Linux install) to use SSL

These notes were developed using Blackboard Learning System (Enterprise) version 6.3.1.374-hf1, running on Red Hat Linux Advanced Server Release 2.1AS/i686. This clean install was initially set to use Blackboard's Challenge-Response authentication mechanism (although later it will be changed to authenticate via Shibboleth).

Part 1: Configuring the Apache Web Server to use SSL

These instructions are essentially the same as those you would use to configure any other Apache server to communicate using SSL, with a few extra changes made to Blackboard's configuration files.

Note

The Apache Web Server must be correctly configured to communicate over SSL before you try to enable SSL Choice from the Blackboard GUI (using the System Administrator Panel) or you will lock yourself out of the server.
The mod_SSL package required to run Blackboard under SSL was not bundled with early releases of Blackboard. Users of versions prior to 6.0.11 will have to follow some additional steps detailed in the Blackboard Setup Guide for UNIX to add this package. Users of versions of Blackboard after 6.3.1 should check the Blackboard Setup Guide for UNIX to ensure that the procedure hasn't changed significantly since these notes were developed.

The following list of points should be followed:

- Obtain a certificate from your chosen Certificate Authority (CA) e.g. GlobalSign. The CA should provide installation instructions, or see the Apache documentation at http://httpd.apache.org/docs/2.0/ssl/ssl_faq.html#aboutcerts [see section 3.1.3]

- The next step is to check that the main Apache configuration file httpd.conf has the necessary flag set to make it use SSL. The Blackboard Setup Guide for UNIX instructs you to add a new line to the file /usr/local/blackboard/apps/httpd/conf/httpd.conf, but on our clean installation, we found it was already present at the very end:

  ```
  Include conf/ssl.conf
  ```

- Blackboard comes with some self signed (and expired!) SSL certificates, you should replace these.

  Your new server key file (server.key) should be placed into this directory: apps/httpd/conf/ssl.key

  Your server certificate file (server.crt) should be stored here: apps/httpd/conf/ssl.crt If you are running 6.3 or 6.2, the Blackboard Setup Guide for UNIX instructs you to inform the server of their location by modifying the bb-config.properties file located at: /blackboard/config/bb-config.properties by adding these two lines:

  ```
  SSLCertificateFile /<path>/server.crt SSLCertificateKeyFile /
  <path>/server.key
  ```

  This instruction appears incorrect. A search of the file showed that two similar entries were already in place:

  ```
  ##########################################################################
   ############## UNIX - Platform Configuration ################
  ##########################################################################


  ##########################################################################
  ```

```
#####################################################################
 ################## ssl configuration settings ##############
#####################################################################

bbconfig.unix.ssl.certificatefile=/usr/local/blackboard/apps/httpd
bbconfig.unix.ssl.certificatekeyfile=/usr/local/blackboard/apps/ht
```

Therefore we did not make any further changes to the bb-config.properties file If you are using an older version of Blackboard the Blackboard Setup Guide for UNIX instructs you to add (or check for the existence of?) similar lines in this file instead:

`/usr/local/blackboard/apps/httpd/conf/ssl.conf`

We expect that this is where you would really have to add these two lines:

`SSLCertificateFile /<path>/server.crt SSLCertificateKeyFile / <path>/server.key`

- Now stop and restart your Blackboard server so that these changes take affect.

  From `/usr/local/blackboard/tools/admin/` issue the command:

  `ServiceController.sh services.webserver.restart`

- You can now test the configuration is correct by going to the https version of your normal blackboard URL: https://blackboard.uni.ac.uk/. If this page opens and then automatically redirects you to the normal (insecure) login page (http://blackboard.uni.ac.uk/) without complaining about certificates then everything is working. Blackboard will not accept login via SSL until you complete the process by enabling SSL from the System Administrator panel (see Section 3.2.2: Part 2).

If you get complaints about unsigned certificates read the section on Certificates signed by an Intermediate below.

Certificates signed by an Intermediate

If your CA has issued a certificate that is itself signed by an intermediate certificate, you will require a chain of one or more certificates to get to one of the root certificates the browser will trust. Otherwise you will see a range of certificate warnings when you point your browser at this URL.

You can use this command to see what authority your certificate is signed by:

`openssl x509 -noout -text -in server.crt`

In our case the certificate was not signed directly by an authority listed in the file: apps/httpd/conf/ssl.crt/ca-bundle.crt Thus we needed to download the intermediate certificate(s) in PEM format from the CA. In the case of the GlobalSign website, we visited the 'GlobalSign Trust Hierarchy' section and downloaded two files: primserver.pem and ServerSign.pem. With these extra files you can either:

- Save these two PEM files to the apps/httpd/conf/ssl.crt/ directory, and append ".crt" to the end of the filenames (e.g. to create primserver.pem.crt and ServerSign.pem.crt)
- Run make in that directory

```
make
```

- Assuming your blackboard installation is at /usr/local/blackboard add this line to the file `apps/httpd/conf/ssl.conf` :

```
SSLCACertificatePath /usr/local/blackboard/apps/httpd/conf/ssl.crt/
```

Note that these changes to apps/httpd/conf/ssl.confb will be overwritten the next time you run `Push-ConfigUpdates.sh` unless you also make them to the file `apps/httpd/conf/ssl.conf.bb`. Or, alternatively:

- Concatenate the two PEM files together to give a single certificate file (eg. called ca.crt)
- Put this file in the `apps/httpd/conf/ssl.crt/` directory
- Assuming your blackboard installation is at /usr/local/blackboard add this line to the file `apps/httpd/conf/ssl.conf` :

```
SSLCertificateChainFile                                    usr/local/blackboard/apps/httpd/conf/ssl.crt/ca.crt
```

Either way, after stopping and restarting the server, this should allow you to go to the https address of your Blackboard site without certificate warnings.

Part 2: Enabling SLL Choice via the Administrator Panel

SSL configuration options are set from the Blackboard System Administration panel. Typically a user must have System Administrator rights to alter this setting. To locate the SSL choice option:

- Log on to your blackboard server with an account which has System Administrator rights
- Click on the System Admin tab to display the Administration Panel
- Under the heading Security, click on the option SSL choice

A standard Blackboard stepped page is shown. The key choice is made in step 2. The three options are essentially: - SSL disabled system-wide - SSL enabled system-wide - SSL enabled only for specified areas (specified in steps 3 - areas, step 4 - tools and step 5 - system extensions).

Tip

The option enable SSL system-wide was selected for maximum security and minimum maintenance. Depending on the configuration of your servers and usage patterns, you might find this gives too great a performance hit. In this case you may want to only enable SSL communication in selective areas. This will require you to review SSL settings when you add new course tools or system extensions (building blocks).

- After you have made your choice, press the Submit button (step 6) to complete the process.
- You should now be able to connect to your Blackboard server using SSL: https://blackboard.uni.ac.uk/.

Collaboration Server

Typically a Blackboard installation includes a second web server which runs the Java-based Virtual Classroom and Chat tools. This server is known as the collaboration server and needs to be separately configured for SSL communication.

As this has no relevance to the Shibboleth project, the reader is referred to the standard configuration options given in chapter 5 of the Blackboard Setup Guide for UNIX . They would also be advised to review recent postings about App Pack 3, SSL and the Collaboration Server on the BBADMIN-L listserver: (http://is.asu.edu/instruction/faq/usingBBADMIN-L.html).

Management

The key thing to watch out for during the installation is which system you will authenticate against, e.g. your local Active Directory Server; this is only an issue as Blackboard may be authenitcating locally, using its internal store of passwords. Note that a Blackboard test server should be used so that you can experiment without messing up the production service. Blackboard is not ready for a production Shibboleth service yet.

Institutional requirements

In this case the institutional managers only need to know that the authentication mechanism is being switched to Shibboleth, and it would be useful for them to understand why this will be required in future. The Blackboard implementation itself should not be altered. Also, the out of the box Blackboard does not support WAYFs.

Existing agreements within the institution should cover local users' access to Balckboard. However, access to Shibboleth-controlled resources available from within the federation you joined may require separate agreements to be set up.

Issues with installation

It is likely that the biggest issues with carrying out the above installation will arise from gaps and lack of clarity as to the necessary steps to be taken in the documentation supplied with Blackboard. For example, in the Shibboleth Integration overview, supplied at http://www.blackboard.com/docs/r7/70/en_US/admin/bbas_r7_0_a... **[NB NEED FULL URL FROM MALCOLM]** under the installation instructions:

“3. Download the correct Shibboleth package for the operating system and install it...”

with no further explanation offered. This has been fed back to Blackboard, and they have responded that for the next version the instructions will be improved.

There are currently no attribute issues as the out of the box Blackboard does not support WAYFs - the Shibboleth Identity Provider server's address has to be written into the code in the configuration file. Blackboard have also been made aware of this.

Deployment of the Zope VLE as a Shibboleth target

to follow

Error Messages and Solutions

The following is a a semi-official list of errors and solutions by the internet2 people: https://umdrive.memphis.edu/wassa/public/shib.faq/shibboleth-faq.html

which IAMSECT developers found useful.

apache

SELinux

```
Cannot load /opt/shibboleth-1.2.1/libexec/mod_shib_20.so into server: /opt/shibboleth-1.2.1/libexec/mod_shib_20.so: failed to map segment from shibded object: Permission denied
```

Apparently this is an SELinux factor. `/var/log/messages` shows:

```
kernel: audit(1115814837.683:0): avc: denied { execute } for pid=8268 comm=httpd path=/opt/shibboleth-1.2.1/libexec/mod_shib_20.so dev=sda1 ino=1016775 scontext=root:system_r:httpd_t tcontext=root:object_r:usr_t tclass=file
```

The solution (which may be temporary) is edit `/etc/selinux/config` change enforce to permissive, and reboot

See http://fedora.redhat.com/docs/selinux-faq-fc3/index.html

mod_ssl

```
...[error] SSL Library Error: 218529960 error:0D0680A8:asn1 encoding routines:ASN1_CHECK_TLEN:wrong tag ...[error] SSL Library Error: 218595386 error:0D07803A:asn1 encoding routines:ASN1_ITEM_EX_D2I:nested asn1 error
```

Take care not to mix up the ‘csr’ (certificate signing request) and ‘crt’ (the signed certificate) in the ssl.conf file. Correcting this and restarting apache solves the problem.

Origin: Invalid assertion consumer service URL

Invalid what consumer what what?!

Logging in at the origin can result in this error to the user:

```
Handle Service failure at (/shibboleth/HS)
```

edu.internet2.middleware.shibboleth.hs.Handle Servlet$Invalid Client Data Exception:

Invalid assertion consumer service URL.

And in the HS’s logs: .

```
... INFO [HS] ... - Handling request.
```

```
... DEBUG [HS] ... - Remote provider has identified itself as: (https://example.org/shibboleth/target). ... DEBUG [HS] ... - Provider is a member of group (https://example.org/shibboleth), but no matching Relying Party was found.
```

```
... INFO [HS] ... - Could not locate Relying Party configuration for
```

```
(https://example.org/shibboleth/target). Using default Relying Party:
(urn:mace:ac.uk:sdss.ac.uk:federation:sdss).

... INFO [HS] ... - Supplied consumer URL not found in metadata.

... ERROR [HS] ... - Supplied assertion consumer service
URL(http://webdev2.ncl.ac.uk/Shibboleth.shire) is NOT valid for pro-
vider (https://example.org/shibboleth/target).

... ERROR [HS] ... -
edu.internet2.middleware.shibboleth.hs.HandleServlet$InvalidClientData
Exception: Invalid assertion consumer service URL.

... DEBUG [HS] ... - servletPath=/hserror.jsp, pathInfo=null, queryS-
tring=null, name=null

... DEBUG [HS] ... - Path Based Forward

... DEBUG [HS] ... - Disabling the response for futher output
```

The solution for this is that the providerId in the target's `shibboleth.xml` is still set to the default, https://example.org/shibboleth/target. You need to either invent a unique URL for it (as per http://shibboleth.internet2.edu/guides/deploy-guide-target.html), or, if you are in a federation, you may have your providerId dictated to you.

Target

You may get the following error:

```
SHIRE failure at (http://webdev2.ncl.ac.uk/Shibboleth.shire)
```

Exception: Fatal Profile Error: SAMLPOSTProfile::getSSOAssertion() unable to start session due to policy mismatch (target policies: urn:mace:inqueue, https://webdev2.ncl.ac.uk/shibboleth/target)

You should set your providerId to a recognised thing.

A. Appendix

Sample shibd init script

Based on http://archie.csce.uark.edu/gpn/target.html.

```
# Shibboleth shibd startup script
#
# chkconfig: - 82 17
# description: shibd is the shibboleth attribute requester
#
# processname: shibd

. /etc/rc.d/init.d/functions
shibd=${shibdD-/opt/shibboleth/bin/shibd}
prog=shibd
RETVAL=0

start() {
        echo -n $"Starting $prog: "
```

```
        $shibd -f $OPTIONS &
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && touch /var/lock/subsys/shibd

        return $RETVAL
}

stop() {
        echo -n $"Stopping $prog: "
        killproc $shibd
        RETVAL=$?
        echo

        [ $RETVAL = 0 ] && rm -f /var/lock/subsys/shibd /var/run/shibd.pid
}

reload() {
        echo -n $"Reloading $prog: "
        check13 || exit 1
        killproc $httpd -HUP
        RETVAL=$?
        echo
}

case "$1" in
        start) start ;;

        stop) stop ;;

        restart)
                stop
                start
        ;;

        *)
                echo $"Usage: $prog {start|stop|restart}"
                exit 1
        ;;
esac

exit $RETVAL
```