Version	Date	Comments
1.0	15 January 2009	Stijn Lievens
1.0.1	2 April 2009	Stijn Lievens. Corrected some typos and mentioned that one also needs to set APACHE_HOME when compiling the mod_permis module.

Installation of the Shibboleth-Apache Authorisation Module

1. Introduction

This guide aims to give a step-by-step guide to the installation of the Shibboleth-Apache Authorisation Module version 5.0.x on a Fedora Linux system.

We assume that the Fedora system has the development tools installed. You should at least have a JDK version 1.5 installed.

The steps involved are the following.
- Obtain and compile the Apache server software.
- Obtain the SAAM software and compile two apache modules.
- Adapt the Apache configuration files to use PERMIS authorization for the protection of certain web pages.

A very common way of working is that the authentication and authorization information is stored in an LDAP server. This guide does **not** intend to explain how to set up and configure the LDAP server. There is an LDAP server available that allows you to test the Apache configuration prior to setting up your own LDAP server.

2. Obtain and compile the Apache server software

Important note: the SAAM software currently only supports versions 1.3 and 2.0 of the Apache web server software. In particular Apache web server 2.2 is not yet supported.

First, download the source code of the web server software.

```
~$$ wget http://mirror.public-internet.co.uk/ftp/apache/httpd/httpd-2.0.63.tar.gz
```

Next, unpack it:

```
~$$ tar -xzvf httpd-2.0.63.tar.gz
```

This will create a directory, called httpd-2.0.63, which holds all the files necessary to compile the web server.

Then, set the APACHE_HOME variable to where you want to install the web server to.

```
~$$ export APACHE_HOME=/usr/local/apache2.0
```

Next, configure your apache installation for compilation, so move into the directory created by the tar command.

```
~$$ cd httpd-2.0.63
httpd-2.0.63$$ ./configure --prefix=$APACHE_HOME --enable-so \
                        --enable-mods-shared=all --enable-proxy
```

Next, we have to 'make' it:

```
httpd-2.0.63$$ make
```

And, finally, as user root (depending on your APACHE_HOME value) do.

```
httpd-2.0.63>> make install
```

3. Obtain the SAAM software and compile the two necessary modules

In order to download PERMIS software you need to register. You can do this by visiting this page: http://sec.cs.kent.ac.uk/permis/essentials/register.shtml

You will then receive username/password combination to download the software:

```
~$$wget --user=the_user_name --password=the_password \
http://sec.cs.kent.ac.uk/permis/private/saam/saam_5_0_5.zip
```

Unzip the file into a directory called 'saam'

```
~$$ unzip saam_5_0_5.zip -d saam
```

3.1 Compile and install the mod_auth_ldap module

Make sure that the APACHE_HOME and LDAP_DIR variables are set:

```
~$$ export APACHE_HOME=/usr/local/apache2.0 ; export LDAP_DIR=/usr
```

Then, inside the saam/mod_auth_ldap3.0 directory do

```
mod_auth_ldap3.0$ ./configure --with-apxs=$APACHE_HOME/bin/apxs --with-ssl=no\
      --with-ldap-dir=$LDAP_DIR --with-apache-dir=$APACHE_HOME \
      --with-apache-ver=2
```

Followed by:

```
mod_auth_ldap3.0$ make
```

As root do:

mod_auth_ldap3.0>> make install

3.2 Compile and install the mod_permis module

In order for the compilation of the mod_permis module to succeed, you need to have a JDK installed on your system (and not just a JRE). In Fedora, you can install a JDK as follows:

~>> yum install java-1.6.0-openjdk.i386
~>> yum install java-1.6.0-openjdk-devel.i386

Enter the saam/mod_permis_src directory:

In order to compile the mod_permis module we have to update the variable LD_LIBRARY_PATH. This is done in reference to the JAVA_HOME variable

mod_permis_src$$ export JAVA_HOME=/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0/
mod_permis_src$$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/jre/lib/i386/client:$JAVA_HOME/jre/lib/i386

Ensure that the APACHE_HOME variable is still set:

mod_permis_src$$ export APACHE_HOME=/usr/local/apache2.0

Next, make sure that the file make20.sh is executable and execute it:

mod_permis_src$$ chmod 755 make20.sh ; ./make20.sh

Note: if the compilation fails saying something like "permisJNI.c:21:17: error: jni.h: No such file or directory" then this probably means that you don't have a JDK installed. In particular, your $JAVA_HOME directory should contain a directory named 'include'.

The next step is to copy the module to the relevant place. As root do:

mod_permis_src>> cp mod_permis_20.so $APACHE_HOME/modules

3.3 Copy jar and configuration files

As root, navigate to the saam directory and copy the jar files it contains to the $JAVA_HOME/jre/lib/ext directory:

*saam>> cp *.jar $JAVA_HOME/jre/lib/ext*

If you would like to try the example configuration provided with the software (recommended for beginners) then also copy the following files

```
saam>> cp policy1.xml rootca.cer $APACHE_HOME/conf
```

4. Adapt the Apache configuration files and try the example

4.1 Adapt the Apache configuration files

As a first step attach the contents of the file httpd-addition.conf to the end of the $APACHE_HOME/conf/httpd.conf file:

```
saam>> cat httpd-addition.conf >> /usr/local/apache2.0/conf/httpd.conf
```

Next, make sure that the line LoadModule mod_permis points to the mod_permis_20.so file on your system (and that this file is readable). You may also have to change the value of the variables PermisPolicyLocation, PermisRootCA and PermisJavaClass to point to the correct files. (Only the path should need changing, the base name should be correct.)

You will also have to create two files (and a directory) so that the different modules can do their logging properly. In the directory /logs do, as root:

```
logs>> touch mod_permis.log permis.log; chmod 666 *.log
```

If you don't do this, then decisions will **not** be made.

4.2 Try it

Create a simple web page and put it in the location $APACHE_HOME/htdocs/secure2/index.html.
Create a second one to put in the $APACHE_HOME/htdocs/public/index.html.
This assumes that you have a default apache installation. See the DocumentRoot variable in the httpd.conf file.

Now the time has come to start the web server. Navigate into the $APACHE_HOME/bin directory and do:

```
bin>> ./apachectl start
```

If you see the error: "libjvm.so: cannot open shared object file: No such file or directory" this probably means that the LD_LIBRARY_PATH variable is not set correctly.

If you find error like “Syntax error on line 1080 of /usr/local/apache2/conf/httpd.conf: Cannot load /usr/local/apache2/modules/mod_permis_20.so into server: /usr/local/apache2/modules/mod_permis_20.so: cannot restore segment prot after reloc: Permission denied” then as a root you have to type

```
~>> chcon -t textrel_shlib_t '/usr/local/apache2/modules/mod_permis_20.so'
```

Visiting the web page http://localhost/ should confirm that the Apache web server is up and running.

Next, visiting http://localhost/public/index.html should show you the page you put there without requiring any authentication.

Finally, we come to the page http://localhost/secure2/index.html. If you authenticate as User0 with password 'User0' then you should get access to the page. However, if you authenticate as User1 with password 'User1' then you will get a 'Authorization required' page, because User1 doesn't have the necessary credentials to access this particular page.

It is important to clear the authenticated sessions between different accesses of the secure page. Otherwise, your browser will think you are still authenticated and will not ask to authenticate again.