InCommon

InCommon Authentication is an unsupported feature of HUBzero. This documentation has not been verified against HUBzero 2.2 or Debian 8+. It is included here for completeness. Please send any corrections or feedback to support@hubzero.org

Introduction

This plugin provides some code necessary to allow your hub to accept credentials using the Shibboleth system. Most commonly, this implies membership in the InCommon network.

Shibboleth has some particular architectural demands, namely that it will install a new daemon and a new Apache module on your system. InCommon has some administrative demands, in that you will need to negotiate to get your hub added to their XML manifest as a service provider.

Installation

Debian

```
# apt-get install -y libapache2-mod-shib2
```

Redhat Enterprise Linux & other distributions

See Shibboleth wiki entry on service provider installation for information on how to add the Shibboleth software to your list of repositories so that it can be installed and upgraded through yum, or, failing that, how to install from SRPMS.

Configuration

Shibboleth

Certificates

As root, run the script shib-keygen, which was installed as part of the package. This will generate a key pair for your service provider to use. No further configuration is required for this; the software will find the keys when the shibd service is restarted.

output

```
Generating a 2048 bit RSA private key
.....................................................................
..................+++
.....................+++
writing new private key to '/etc/shibboleth/sp-key.pem'
-----
```

/etc/shibboleth/attribute-map.xml

This file controls which attributes (bits of user information) the software will extract during login when the identity provider makes them available.

Make sure the following pertinent attributes are not commented out in both forms of the “name” attribute.

eppn (username, probably already enabled in the shipped configuration):

```
    <Attribute name="urn:mace:dir:attribute-
def:eduPersonPrincipalName" id="eppn">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
    </Attribute>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" id="eppn">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
    </Attribute>
```

Name & email (probably not enabled by default):

```
    <Attribute name="urn:mace:dir:attribute-def:sn" id="sn"/>
    <Attribute name="urn:mace:dir:attribute-
def:givenName" id="givenName"/>
    <Attribute name="urn:mace:dir:attribute-
def:displayName" id="displayName"/>
    <Attribute name="urn:mace:dir:attribute-def:mail" id="mail"/>
    <Attribute name="urn:oid:2.5.4.4" id="sn"/>
    <Attribute name="urn:oid:2.5.4.42" id="givenName"/>
    <Attribute name="urn:oid:2.16.840.1.113730.3.1.241" id="displayNam
e"/>
    <Attribute name="urn:oid:0.9.2342.19200300.100.1.3" id="mail"/>
```

/etc/shibboleth/shibboleth2.xml

This is the main configuration, which controls how the software federates with identity providers.

First, replace $YOUR_HOSTNAME with, uh, your hostname, in the entityID attribute near the top of the file:

```
       <ApplicationDefaults entityID="https://$YOUR_HOSTNAME/login/shi
bboleth" REMOTE_USER="eppn persistent-id targeted-id">
```

In the block, delete or comment-out any SSO or SessionInitiator blocks that shipped, and add the two listed below, again interpolating your real hostname. This tells the software to check with the HUBzero CMS plugin about where to redirect for a given authentication request, and allows the HUBzero CMS to selectively enable providers.

```
        <Sessions lifetime="28800" timeout="3600" relayState="ss:mem"
                  checkAddress="true" handlerSSL="true" cookieProps="h
ttps">
            <SSO discoveryProtocol="SAMLDS" ECP="true" discoveryURL="h
ttps://$YOUR_HOSTNAME/login?authenticator=shibboleth&wayf">
                   SAML2 SAML1
            </SSO>
            <SessionInitiator type="Chaining" Location="/login/shibbol
eth" isDefault="true" id="Login">
                <SessionInitiator type="SAML2" template="bindingTempla
te.html"/>
                <SessionInitiator type="Shib1"/>
                <SessionInitiator type="SAMLDS" URL="https://$YOUR_HOS
TNAME/login?authenticator=shibboleth&wayf"/>
            </SessionInitiator>
            <!-- Default <Handler> tags not pictured, but they should
stay -->
       </Sessions>
```

If you run into issues where you seem to be stuck in a redirect loop between the idp and the sp, changing the cookie properties to use a less specific path may help.

```
cookieProps="; path=/; secure; HttpOnly"
```

If this is a production machine you will want to set a real email for the support contact:

```
        <Errors supportContact="support@$YOUR_HOSTNAME"
             helpLocation="/about.html"
             styleSheet="/shibboleth-sp/main.css"/>
```

Finally, you will need to configure how and where the software looks for metadata about identity providers. This is just a list of providers you can support, including some helpful annotations like where the service URLs and what public key to use when communicating with it.

Metadata provider: TestShib

For development and test machines it is often useful to use TestShib, and its configuration looks like this, below the Sessions tag and at the same scope:

```
      <MetadataProvider type="XML" uri="http://www.testshib.org/metada
ta/testshib-providers.xml" backingFilePath="testshib-two-idp-
metadata.xml" reloadInterval="180000"/>
```

Visit the TestShib site for more information about how to set this up, if you're interested. Hopefully you do not need the "Install" selection, but pick up from "Register". During "Configure" it recommends replacing your whole shibboleth2.xml with one it generated. Make a backup if you do, or else just add the MetadataProvider above to your existing configuration.

When you reach "Test", see below for the HUBzero CMS configuration that will add TestShib to the list of available identity providers on your hub.

Metadata provider: InCommon

If your plans include membership in the InCommon consortium, this is the incantation, below the Sessions tag and at the same scope:

```
        <MetadataProvider type="XML" uri="https://wayf.incommonfederat
ion.org/InCommon/InCommon-metadata.xml" backingFilePath="federation-
metadata.xml" reloadInterval="7200">
            <MetadataFilter type="RequireValidUntil" maxValidityInterv
al="2419200"/>
            <MetadataFilter type="Signature" certificate="inc-md-
cert.pem"/>
        </MetadataProvider>
```

Install https://ds.incommon.org/certs/inc-md-cert.pem as /etc/shibboleth/inc-md-cert.pem so it's available for this provider.

Metadata provider: others?

If you are doing one-on-one negotiations with identity providers the metadata situation gets a bit more hairy, but the identity providers in question will probably be able to guide your configuration.

Apache

Quoth the Shibboleth wiki entry on service provider installation:

- UseCanonicalName On
- Ensure that the ServerName directive is properly set, and that Apache is being started with SSL enabled.

Make sure installing the software enabled both the module shib2 and the support daemon shibd.

Typically this means that there is a symlink /etc/apache2/mods-enabled/shib2.load that points to /etc/apache2/mods-available/shib2.load and that this report works:

```
# service shibd status
[ ok ] shibd is running.
```

/etc/apache2/sites-enabled/{your-ssl-enabled-config-file}

Your EntityID is something like https://hostname/login/shibboleth, but the actual URL to pick up the login process again in HUBzero CMS terms is more complicated, so we rewrite it. I recommend putting this statement as high as possible in the config (after RewriteEngine on) so that the "L"ast last triggers and you can be assured the URL is not subsequently rewritten by anything else you're doing.

```
        RewriteCond     %{REQUEST_URI}          ^/login/shibboleth
        RewriteRule     (.*) /index.php?option=com_users&authenticator
```

```
=shibboleth&task=user.login [NC,L]
```

Bind an endpoint to the module. This is used during the login process and is also useful to get a basis for your service provider's metadata, which is served at /Shibboleth.sso/Metadata when the request comes from localhost.

```
        <Location /Shibboleth.sso>
                SetHandler shib
        </Location>
```

You probably have a rule that directs all requests that appear to be for HUBzero CMS content to the index.php bootstrap, and we need to note that /Shibboleth.sso isn't HUBzero CMS business, so make sure you have a RewriteCond like this:

```
        RewriteCond     %{REQUEST_URI}          !^/Shibboleth.sso/.*$
[NC]
        RewriteRule     (.*)                    index.php
```

Finally, we actually protect the entityID location /login/shibboleth. We can redirect a user to this path to require them to make a Shibboleth login. Shibboleth won't know specifically how to do that so it will make a request to the wayf location defined above in shibboleth2.xml. This is part of the HUBzero CMS that knows already which provider the user selected from the login page, so it spits back the appropriate identity provider entityId. From there the metadata is referenced to find the endpoint associated with that institution, and the user is sent to the login page. They come back to /login/shibboleth upon submission, but now the requirement to have a Shibboleth session is satisified, and the rewritten URL referencing user.login is served to complete the process.

```
        <Location /login/shibboleth>
                AuthType shibboleth
                ShibRequestSetting requireSession 1
                Require valid-user
        </Location>
```

Restart the shibd and apache2 services when satisified with this configuration.

HUBzero CMS

Plugin

Log in to /administrator, choose Extensions and then Plugin Manager, and locate the Shibboleth plugin in the Authentication category.

If you would like to selectively hide the Shibboleth login options for testing, enter something in the “Testing mode key” field. This term must appear in the query string for the controls of the plugin to render. For example, if you enter “incommon” you should test the login page by visiting “/login?reset=1&incommon” (reset=1 in case it remembers your logging in with a different method, in which case you’ll only see the controls for that method anyway).

The links section is useful only for testing, but it can be used to destroy a link between your test account and a particular institution so that you can try it again.

The Institutions section is where you manage which options are presented on the login page.

An example of an entry here, for the TestShib public identity provider test mechanism described above:

```
Entity ID: https://idp.testshib.org/idp/shibboleth
Label: TestShib
Host: testshib.org
```

The entity ID must strictly match what you have in your metadata provider, but the label is free-form and the host is optional. The login page attempts to do a reverse-DNS of the user’s IP to see if they are on a particular network. If it turned out in this case that the client was from *.testshib.org this option would be pre-selected in the plugin’s controls.

Save your settings with the button near the top right of the page when you’re done.