Simple Guide to Configuration Files

Christopher Simpson
Swansea College

Table of Contents

These guides have been prepared by organisations who participated in the JANET Shibboleth on Windows project. These guides are provided for general information purposes and are not intended to be definitive or exhaustive guides to the configuration, installation and implementation of Shibboleth On Windows.

Configuration Files

Generic Information

Shibboleth configuration files are all XML based. These files must be correctly structured as this is the most common cause of Shibboleth failure.

The Shibboleth configuration files are located in:

C:\Program Files\Internet2\IdP\etc

The Shibboleth configuration files we will be looking at in more detail are:

- idp.xml
- resolver.xml
- arps\arp.site.xml

with a very brief look at:

- metadata.xml

Idp.xml

This is the generic configuration file for the Shibboleth IDP service. If you use the quick installer then there are not many / any changes you will need to make here.

IdPConfig

```
<IdPConfig
        xmlns="urn:mace:shibboleth:idp:config:1.0"
        xmlns:cred="urn:mace:shibboleth:credentials:1.0"
        xmlns:name="urn:mace:shibboleth:namemapper:1.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:mace:shibboleth:idp:config:1.0 ../schemas/shibboleth-idpconfig-
1.0.xsd"
        AAUrl="https://$IDP_ADDR:8443/shibboleth-idp/AA"
        resolverConfig="file:/C:/Program%20Files/Internet2/idp//etc/resolver.xml"
        defaultRelyingParty="urn:mace:shibboleth:examples"
        providerId="https://shib.swancoll.ac.uk/shibboleth">
```

The only tag you are likely to amend here is the tag that defines the location and name of the resolver file:

```
        resolverConfig="file:/C:/Program%20Files/Internet2/idp//etc/resolver.xml"
```

Logging

```
<Logging>
        <ErrorLog level="DEBUG" location=" file:/C:/Program%20Files/Internet2/idp//logs/shib-
error.log" />
        <TransactionLog level="DEBUG"
location="file:/C:/Program%20Files/Internet2/idp//logs/shib-access.log/>
</Logging>
```

The logging tag defines the level and location of the logs. The suggested level whilst in the testing phase is DEBUG, but INFO is sufficient after test phase is complete. The DEBUG option does create very detailed information regarding the process and consequently large log files can be created.

Credentials

```
<Credentials xmlns="urn:mace:shibboleth:credentials:1.0">
        <FileResolver Id="example_cred">
                <Key>
                <Path>file:/C:/Program%20Files/Internet2/idp/etc/shib.swancoll.ac.uk.key</Path>
                </Key>
                <Certificate>
                <Path>file:/C:/Program%20Files/Internet2/idp/etc/shib.swancoll.ac.uk.crt</Path>
                </Certificate>
        </FileResolver>
</Credentials>
```

The credential tag defines the location of the certificates to be used to encrypt data being sent between the IdP and SP servers.

MetadataProvider

```
        <MetadataProvider
type="edu.internet2.middleware.shibboleth.metadata.provider.XMLMetadata"
            uri="file:/C:/Program%20Files/Internet2/idp/etc/UKmetadata.xml"/>
```

The metadata tag determines the metadata file to be used. The uri is the setting that you are most likely to change. The full path should be entered here. Details of the metadata file can be found further into this document.

Resolver.xml

Description

This file determines where attributes are located and which attributes can be retrieved. It is possible to use multiple sources, i.e. databases and LDAP Implementations.

The below example is a LDAP connection example. This connects to our sample Active Directory (Microsoft's implementation of LDAP)

```
<JNDIDirectoryDataConnector id="directory">
                        <Search filter="sAMAccountName=%PRINCIPAL%">
                                <Controls searchScope="SUBTREE_SCOPE" returningObjects="true" />
                        </Search>
                        <Property name="java.naming.factory.initial" value="com.sun.jndi.ldap.LdapCtxFactory" />
                        <Property name="java.naming.provider.url"
value="ldap://10.10.1.99:389/ou=Swansea%20College,dc=SWANCOLL,dc=AC,dc=UK" />
                        <Property name="java.naming.security.principal"
value="cn=ouruser,cn=Users,dc=SWANCOLL,dc=AC,dc=UK" />
                        <Property name="java.naming.security.credentials" value="ourpassword" />
</JNDIDirectoryDataConnector>
```

Key elements are:

<Search filter="sAMAccountName" …..

sAMAccountName is used by Microsoft's implementation of LDAP to store the username. Other implementations may use different elements.

<Controls searchScope="SUBTREE_SCOPE" />

By selecting SUBTREE_SCOPE all accounts in sub containers will be included in the search, i.e. for example if you had nested containers Staff and Students within a container CollegeUsers, you could point the provider.url at CollegeUsers and entries in CollegeUsers, Staff and Students would all be searched. This can be very useful!

<Property name="java.naming.provider.url" ……

This is the provider address. This is the IP address of the server that holds your Active Directory. Usually port 389 is used. The full address of the root OU follows. Please not %20 is used instead of a space.

<Property name="java.naming.security.principal" ……

This contains the location of the user with permissions to query the Active Directory. Remember that you must give this user the

adequate permissions. This is the full name as can be seen from the example above.

<Property name="java.naming.security.credentials" …….

This contains the password of the user defined above.

Attributes that are allowed to be released are also defined in the resolver:

```
<SimpleAttributeDefinition id="urn:mace:dir:attribute-def:sAMAccountName">
        <DataConnectorDependency requires="directory"/>
</SimpleAttributeDefinition>
```

The above example allows the sAMAccountName to be used in conjunction with the ARP file in making this information available to the service providers (SPs).

Also note that the above states that this information should come from "directory" which is the Active Directory definition above.

We could have multiple sources of data, i.e. the username coming from LDAP and other information coming from a Database (i.e. course enrolments).

The resolver can be as simple or complex as you like.

Note that the ARP (Attribute Release Policy) restricts which SPs can access the different information sources.

Arp.site.xml

```
<Rule>
        <Target>
                <Requester>https://sp.swancoll.ac.uk/shibboleth-sp</Requester>
        </Target>

        <Attribute name="urn:mace:dir:attribute-def:sn">
                <AnyValue release="permit"/>
        </Attribute>

        <Attribute name="urn:mace:dir:attribute-def:givenName">
                <AnyValue release="permit"/>
        </Attribute>

        <Attribute name="urn:mace:dir:attribute-def:mail">
                <AnyValue release="permit"/>
        </Attribute>

        <Attribute name="urn:mace:dir:attribute-def:sAMAccountName">
                <AnyValue release="permit"/>
        </Attribute>

        <Attribute name="urn:mace:dir:attribute-def:cn">
                <AnyValue release="permit"/>
        </Attribute>
</Rule>
```

The ARP is an xml document that determines which attributes can be released to the various Service Providers. There can be multiple rules in an ARP. Each rule must have a requester which is the Service Provider URL.

Multiple Attributes can be defined, but every entry must already be defined in the resolver.xml file, for example:

The entry in resolver.xml

```
<SimpleAttributeDefinition id="urn:mace:dir:attribute-def:mail">
        <DataConnectorDependency requires="directory"/>
</SimpleAttributeDefinition>
```

relates to this entry in the arp.site.xml

```
<Attribute name="urn:mace:dir:attribute-def:mail">
        <AnyValue release="permit"/>
</Attribute>
```

Metadata.xml

The metadata file contains details of all the other Service Provider and Identity Providers within the federation.

These are usually supplied to you when you join a federation, i.e. testshib so you shouldn't need to amend anything in here.

The metadata.xml file is generally made up of EntityDescriptors. There is one format for each IdP and a slightly differing format for the SPs.

Here is the entry for the Swansea College IdP server within the UK Access Management Federation:

```
<EntityDescriptor ID="uk000366" entityID="https://shib-idp.swancoll.ac.uk/shibboleth">
    <!--
        This is an IdP for Swansea College.
    -->
    <Extensions>
        <shibmeta:Scope xmlns:shibmeta="urn:mace:shibboleth:metadata:1.0"
regexp="false">swancoll.ac.uk</shibmeta:Scope>
        <AccountableUsers
xmlns="http://ukfederation.org.uk/2006/11/label"></AccountableUsers>
        <UKFederationMember
xmlns="http://ukfederation.org.uk/2006/11/label"></UKFederationMember>
    </Extensions>
    <IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:1.1:protocol
urn:mace:shibboleth:1.0">
        <Extensions>
            <shibmeta:Scope xmlns:shibmeta="urn:mace:shibboleth:metadata:1.0"
regexp="false">swancoll.ac.uk</shibmeta:Scope>
        </Extensions>
        <KeyDescriptor use="signing">
            <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:KeyName>shib-idp.swancoll.ac.uk</ds:KeyName>
            </ds:KeyInfo>
        </KeyDescriptor>
        <ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-
binding" Location="https://shib-idp.swancoll.ac.uk/shibboleth-idp/Artifact"
index="1"></ArtifactResolutionService>
        <NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>
        <SingleSignOnService Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"
Location="https://shib-idp.swancoll.ac.uk/shibboleth-idp/SSO"></SingleSignOnService>
    </IDPSSODescriptor>
    <AttributeAuthorityDescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:1.1:protocol">
        <Extensions>
            <shibmeta:Scope xmlns:shibmeta="urn:mace:shibboleth:metadata:1.0"
regexp="false">swancoll.ac.uk</shibmeta:Scope>
        </Extensions>
        <KeyDescriptor use="signing">
            <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:KeyName>shib-idp.swancoll.ac.uk</ds:KeyName>
            </ds:KeyInfo>
        </KeyDescriptor>
        <AttributeService Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
Location="https://shib-idp.swancoll.ac.uk/shibboleth-idp/AA"></AttributeService>
        <NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>
    </AttributeAuthorityDescriptor>
    <Organization>
        <OrganizationName xml:lang="en">Swansea College</OrganizationName>
        <OrganizationDisplayName xml:lang="en">Swansea
College</OrganizationDisplayName>
        <OrganizationURL xml:lang="en">http://www.swancoll.ac.uk/</OrganizationURL>
```

```
    </Organization>
    <ContactPerson contactType="support">
        <GivenName>IT Helpdesk</GivenName>
        <EmailAddress>mailto:helpdesk@swancoll.ac.uk</EmailAddress>
    </ContactPerson>
</EntityDescriptor>
```

The completed document contains hundreds of such entries and is approximately 1.8Mbytes in size. The metadata file is supplied by the federation upon being accepted into it.

The XML entry above contains identification information of the specific IdP server and also includes contact details for the organisation.

Disclaimer

This guide is provided by JANET(UK) for general information purposes only and the JNT Association cannot accept any responsibility and shall not be liable for any loss or damage which may result from reliance on the information provided in it.