MsADFSIntegration

Integration with Microsoft ADFS

Microsoft's ADFS (Active Directory Federation Services) product provides web-based single sign on capabilities by implementing the "WS-Federation: Passive Requestor Interoperability Profile" specification. The Shibboleth IdentityProvider and ServiceProvider both offer integration with ADFS deployments.

ADFS IdentityProvider Setup

Version 1.3 of the IdentityProvider requires the use of an extension module in order to integrate with ADFS. The latest version of the can be downloaded from http://www.shibboleth.net/downloads/identity-provider/extensions/archive/ . The easiest way to get ADFS integration working is to start with a fully-functional IdP. Once you have successfully tested your IdP, the following additional steps are required:

Install the ADFS extension

- Download the IdP ADFS extension module
- Unpack the extension file into $IDP_HOME/custom/
- Configure an ADFS endpoint in the web application deployment descriptor (Add the following snippet to /webAppConfig/dist.idp.xml)

```
<servlet-mapping>
         <servlet-name>IdP</servlet-name>
         <url-pattern>/ADFS</url-pattern>
</servlet-mapping>
```

- Run "./ant compile" from $IDP_HOME in order to compile the extension
- Run "./ant install" from $IDP_HOME in order to add the extension to your existing IdP installation

Configure the ADFS extension

- Add an ADFS protocol handler to idp.xml

```
<ProtocolHandler implementation="edu.internet2.middleware.shibboleth.idp.provider.ADFS_SSOHandler">
        <Location>https?://[^:/]+(:443)?/shibboleth-idp/ADFS</Location>
</ProtocolHandler>
```

- Add a NameMapping configuration to idp.xml in order to add support for MS UPNs. Reference this mapping from the appropriate `<RelyingParty>` element.

```
<NameMapping
        xmlns="urn:mace:shibboleth:namemapper:1.0"
        id="shm"
        format="http://schemas.xmlsoap.org/claims/UPN"
        class="edu.internet2.middleware.shibboleth.common.provider.UPNNameIdentifierMapping"
        handleTTL="28800" scope="example.org"/>
```

- Add the XML attribute `namespace="http://schemas.xmlsoap.org/claims"` to attribute definitions in resolver.xml for any attributes that should be sent to ADFS providers

- Configure metadata in accordance with recommendations from ADFSMetadataProfile

ADFS ServiceProvider Setup

The 1.3c patch release of ShibOnedotThree includes an optional extension library, `adfs.so` , that provides protocol support for interoperating with the Microsoft ADFS product, included in Windows Server 2003 R2.

To enable support for ADFS-compliant IdPs, an SP must be running 1.3c or later and take the following additional steps:

ShibbolethXml Changes

Inside the top-level `<Extensions>` element, add the following, substituting the appropriate path to the Shibboleth libexec files based on your installation:

```
<Library path="C:/opt/shibboleth-sp/libexec/adfs.so" fatal="true"/>
```

Inside the application section that you want to ADFS-enable, or in the top-level default, add the following to the `<Sessions>` element after the existing `<md:AssertionConsumerService>` :

```
<md:AssertionConsumerService Location="/ADFS" index="4"
    Binding="http://schemas.xmlsoap.org/ws/2003/07/secext" ResponseLocation="/"/>
```

Make sure the `index` value is unique among the set. `ResponseLocation` is used during an ADFS-initiated single logout and specifies where to send the browser after terminating the session.

The `Location` value is up to you, but must be defined in the Resource Partner configuration to the ADFS servers that will be supporting this SP. As with other endpoints, the full endpoint location is determined by appending this value to the `handlerURL` attribute and usually the server hostname. A typical value will be `https://hostname.example.org/Shibboleth.sso/ADFS`

Finally (and this is the really hard part) you have to understand how SessionInitiators work at a fairly deep level if you want to deploy this for real. To test ADFS, the simplest thing to do is to treat this like a bilateral installation with a single IdP partner site. You modify the SessionInitiator element with the `isDefault` attribute to use ADFS and redirect directly to the ADFS Account Partner. To do this, you change the `wayfBinding` attribute to `http://schemas.xmlsoap.org/ws/2003/07/secext` and set the `wayfURL` attribute to the location of the ADFS Account Partner's federation service URL. After these changes, resources that are protected in a default manner will cause an ADFS-style redirect to the `wayfURL` location.

A more complex multi-protocol deployment is harder and depends on the kinds of resources being protected, but is much easier when using the LazySession approach (but this only works for dynamic content).

AttributeAcceptancePolicy Changes

ADFS refers to SAML attributes as *claims*. Various claims are predefined for communicating things like usernames, groups, email addresses. Custom claims are also supported. All claims have a common `AttributeNamespace` of `http://schemas.xmlsoap.org/claims`

Since this is not the default namespace used by Shibboleth, each `<AttributeRule>` element must include a `Namespace` attribute containing the ADFS namespace above.

As an example, the Microsoft-defined Group attribute might be defined using a rule such as:

```
<AttributeRule Name="Group" Namespace="http://schemas.xmlsoap.org/claims" CaseSensitive="true"
                  Header="Shib-Group" Alias="shib-group">
         <AnySite>
                  <AnyValue/>
         </AnySite>
</AttributeRule>
```

The subject of the assertions from an ADFS IdP will typically use a Microsoft-defined format called a UPN that contains the AD username. To export this value to REMOTE_USER, a rule such as the following can be used:

```
<AttributeRule Name="http://schemas.xmlsoap.org/claims/UPN" Header="REMOTE_USER" Alias="user">
         <AnySite>
                  <AnyValue/>
         </AnySite>
</AttributeRule>
```

ADFSMetadataProfile

A Shibboleth SP must be provisioned with SAML Metadata that identifies ADFS partner sites just as Shibboleth and SAML IdPs are defined to it. Refer to the ADFSMetadataProfile for information on how to define partner sites to Shibboleth software.

ADFS Configuration

ADFS requires that each potential SP be registered with it as a Resource Partner. When defining a Resource Partner in ADFS corresponding to an SP, the Federation Service URI and endpoint URL can be extracted from the SP's `entityID` and ADFS-enabled `<AssertionConsumerService> Location` attributes respectively.