NativeSPServiceSSO

The <SSO> element is used to enable and configure support for Single Sign-On/Authentication protocols within the SP. This is of course the primary function of the software, so it is generally present within the <Sessions> element to enable and control SSO settings. It replaces the functions of the <SessionInitiator> and <md:AssertionConsumerService> handler elements from the older (pre-2.4) configuration.

Instead of defining explicit endpoints with low-level binding information, the <SSO> element automates the installation of the appropriate handlers based on the protocols selected for activation. For each protocol activated in the <SSO> element, the order of the Bindings is controlled in `protocols.xml`. Most of the remaining settings are equivalent to the settings supported by the "common" <SessionInitiator> types. Note that the <SSO> element supports many of the attributes of <SessionInitiator>, so some of the default settings can be modified without having to define an entire <SessionInitiator>.

The use of the <SSO> element results in a basic chain of initiator plugins installed at the recommended `"/Login"` handler location. For advanced scenarios that require additional plugins or options, additional explicit <SessionInitiator> elements can be added to the end of the surrounding <Sessions> element. To prevent unforeseen interactions, you may want to remove the shorthand element entirely.

Examples

A basic example using a single, fixed IdP, supporting the usual common SAML protocols:

```
<SSO entityID="https://idp.example.org/idp/shibboleth">
  SAML2 SAML1
</SSO>
```

An example using a SAML Discovery Service and supporting ECP:

```
<SSO discoveryProtocol="SAMLDS" ECP="true" discoveryURL="https://examplefederation.org/DS">
  SAML2 SAML1
</SSO>
```

For a legacy Shibboleth WAYF Service, just replace the `discoveryProtocol` value with `"WAYF"`.

Attributes

- `entityID` (URI)
  - If set, establishes an assumed IdP to use for authentication, if none is passed explicitly with a query string parameter or overridden via content settings.
- `discoveryProtocol` (string)
  - Protocol to use for the Discovery Service, by way of a type of session initiator plugin. Typically either "SAMLDS" (SAML Discovery Service protocol) or "WAYF" (legacy Shibboleth WAYF protocol).
- `discoveryURL` (URL)
  - Location of the discovery service, e.g., https://ds.example.org/DS
- `relayState` (string)
  - Overrides `relayState` setting from the <Sessions> element.
- `entityIDParam` (string)
  - Optional, advanced setting for overriding the name of the query string parameter used to override the IdP to use. Normally `"entityID"` and `"providerId"` are the parameter names supported. This is provided for supporting unusual application requirements.
- `target` (URL)
  - Allows the resources to return to after SSO to be "locked" to a specific value, even when running as a result of active protection of other resources. In other words, this value overrides the actual resource location when SSO redirection is automatic, including initial access and after a timeout.

Other attributes supported include settings specific to various types of <SessionInitiator> plugins to alter the behavior of specific protocols. For example, as above, you can enable ECP support by adding the ECP flag, supported by the SAML2 initiator plugin.

Element Content

The content of the element is a whitespace-delimited list of "protocol" identifiers. Protocol identifiers are listed in preferential order, with the most preferred first. The following are built-in to the SP:

- SAML2
  - SAML 2.0 Browser SSO profile
- SAML1
  - SAML 1.x Browser-POST and Browser-Artifact profiles

An additional protocol is supported if the relevent extension is loaded:

- ADFS

- WS-Federation Passive Interoperability Profile (legacy ADFS)

Other protocols can be "integrated" with the service-based configuration mechanism by supplying the relevant information via the `<ProtocolProvider>` plugin interface.