IdP Support for SAML2

> **Deprecated**
>
> Note that this page has been deprecated. The information it contains is no longer current.

This page shows how to set up an IdP deployment for SAML V2.0 Web Browser SSO. Here you will find procedures for new IdPs as well as existing IdPs migrating from SAML V1.1 to SAML V2.0. We assume that your IdP software has the ability to consume SAML V2.0 requests and issue SAML V2.0 responses.

Generally speaking, the IdP software is configured for SAML V2.0 **before** the IdP's metadata is updated. Unless (and until) SAML V2.0 endpoints appear in IdP metadata, there is no obvious way for an SP to issue a SAML V2.0 authentication request to your IdP. We will exploit this fact to show how to migrate an IdP to SAML V2.0 Web Browser SSO gradually, without abruptly opening the doors to arbitrary SAML V2.0 SPs.

> **Read this entire wiki page before introducing a SAML2 endpoint into metadata!**
>
> An IdP should NOT introduce a SAML2 endpoint into its metadata until it is sure that all of its SP partners are properly configured for SAML2. Most importantly, SP software that is configured for SAML2 but does not have a corresponding SAML2 endpoint in its metadata will cause an error at the IdP.
>
> For example, consider the Shibboleth SP, which is configured for SAML2 SSO out-of-the-box. As long as there are SAML2 SSO endpoints in SP metadata, all is well. Otherwise the following situation can occur: Suppose an IdP partner introduces SAML2 SSO endpoints into its metadata for the first time. The SP, being configured for SAML2 SSO by default, will automatically send a SAML2 request to the IdP (whereas before the SP always sent a SAML1 request). Although the IdP is willing and able to respond to SAML2, the IdP immediately fails since the SP has no SAML2 SSO endpoints in its metadata.
>
> The same is true of SAML2 SLO. An IdP that introduces a SAML2 SLO endpoint into its metadata is inviting an SP to send a logout request. If an SP does so but does not have a SAML2 SLO endpoint in metadata, an error will occur…at the IdP no less!

Configuring the IdP for SAML2 SSO

This section shows how to configure the IdP software for SAML V2.0 Web Browser SSO and how to update metadata. The procedure applies to new IdPs as well as existing IdPs migrating from SAML V1.1 to SAML V2.0.

Preconditions:

- The organization responsible for the IdP is an InCommon Federation participant
- The IdP software supports SAML V2.0 Web Browser SSO

Procedure:

1. Configure the software for SAML V2.0 Web Browser SSO
    - Configure the software with a signing key
    - Configure the software to consume SAML V2.0 authentication requests
    - Configure the software to issue SAML V2.0 assertion responses
2. Update InCommon metadata for SAML V2.0
    - Add the corresponding verification key to metadata
    - Add one or more SAML V2.0 endpoints to metadata

Procedural details:

Configure the IdP software for SAML V2.0 Web Browser SSO at step 1. Most implementations are so configured out-of-the-box. Some implementations even come pre-configured with a (self-signed) signing key. Check your documentation for details.

New IdPs should continue with step 2 and then test the deployment as recommended in the software documentation. New IdPs may also complete the test procedure outlined in the next section, methodically exercising their support of each and every outbound SAML V2.0 binding.

Existing IdPs in production should **not** continue with step 2 above but should instead complete the procedure in the next section. Once that's done (and any uncovered bugs are fixed), the IdP's metadata may be fully updated as indicated in step 2 above.

Migrating the IdP to SAML2 SSO

This section shows how to safely migrate a production IdP deployment to support SAML V2.0 Web Browser SSO.

We assume the IdP deployment is currently issuing SAML V1.1 assertions and has the ability to issue SAML V2.0 assertions. Furthermore, we assume the IdP software supports IdP-initiated SAML V2.0 Web Browser SSO. (Shibboleth IdP 2.3.0 and later satisfy these preconditions, as does simpleSAMLphp. For ideas how to upgrade Shibboleth 1.x to Shibboleth 2.x, visit the Shibboleth wiki.) If so, we can leverage that capability to migrate to SAML V2.0 gradually, without abruptly opening the doors to arbitrary SAML V2.0 SPs.

A flow involving an unsolicited SAML response is called *IdP-initiated SAML Web Browser SSO*. For SAML V1.1 Web Browser SSO, this is the only specified mode of operation in fact. With the introduction of SAML V2.0, SP-initiated flows became commonplace, and in fact IdP-initiated flows were all but forgotten. Here we leverage this quiescent SAML feature to facilitate an IdP's orderly migration to SAML V2.0.

Preconditions:

- The IdP deployment is currently in production
- The IdP deployment is currently issuing SAML V1.1 assertions
- The IdP software supports both SAML V1.1 and SAML V2.0 Web Browser SSO
- The IdP software supports IdP-initiated SAML V2.0 Web Browser SSO

Procedure:

1. Configure an endpoint that supports IdP-initiated SAML V2.0 Web Browser SSO
2. Add a SAML V2.0 `<md:ArtifactResolutionService>` endpoint to IdP metadata
3. Trigger an unsolicited response from the previously configured SAML V2.0 endpoint

Procedural details:

The IdP software is configured for SAML V2.0 Web Browser SSO at step 1. (It's possible that your software was configured for SAML V2.0 out of the box, but since your metadata does not include an `<md:SingleSignOnService>` endpoint with a SAML V2.0 binding, there is no obvious way for an SP to issue a SAML V2.0 authentication request to your IdP.) We will trigger this SAML V2.0 endpoint to issue an unsolicited SAML V2.0 response to an arbitrary SAML V2.0 SP at step 3.

At step 2, a superfluous `<md:ArtifactResolutionService>` endpoint that supports the SAML V2.0 SOAP binding is added to IdP metadata (to work around a limitation of the metadata administrative interface). The appearance of this endpoint in metadata triggers the administrative interface to add the SAML V2.0 protocol URI to metadata, which is enough to let the IdP push a SAML V2.0 response to an SP.

Since the method to trigger an unsolicited SAML V2.0 response is software-specific, step 3 depends on your particular software implementation. See the next section for specific suggestions.

Testing the IdP for SAML2 SSO

If you are using the Shibboleth IdP software, a mechanism to elicit an unsolicited SAML V2.0 response is well documented in the Shibboleth wiki. For example, the following HTTP request triggers an unsolicited response from IdP (urn:mace:incommon:idp.protectnetwork.org) listed in InCommon metadata, targeting the default `<md:AssertionConsumerService>` endpoint at the indicated SP:

```
https://idp.protectnetwork.org/protectnetwork-idp/profile/SAML2/Unsolicited/SSO?providerId=https%3A%2F%2Ffm.
incommon.org%2Fsp
```

In the URL above, note that the entityID of the SP (https://service1.internet2.edu/shibboleth) is suitably encoded.

The previous HTTP request exercises the IdP's ability to respond via a particular SAML V2.0 binding, namely the binding associated with the *default* `<md:AssertionConsumerService>` endpoint at the SP. To exercise other bindings, issue similar requests for other SP endpoints by appending a `shire` parameter to the query string:

```
...&shire=https%3A%2F%2Fservice1.internet2.edu%2FShibboleth.sso%2FSAML2%2FPOST
```

Continue to test the IdP by triggering a response to each endpoint at the SP (of which there are several).

Note that the HTTP parameters used to trigger an unsolicited response (`providerId` and `shire`) are the same parameters used in a Shibboleth 1.x `AuthnRequest`, but since the endpoint at the IdP is a SAML V2.0 endpoint, a SAML V2.0 flow is initiated.

The simpleSAMLphp IdP also supports IdP-initiated login and therefore IdP-first flows are possible with simpleSAMLphp as well. For other implementations, consult your documentation.

External References

- https://wiki.shibboleth.net/confluence/display/SHIB2/IdPUpgrades