SP Support for SAML2

This page shows how to set up an SP deployment for SAML V2.0 Web Browser SSO. The procedures apply to new SPs as well as existing SPs migrating from SAML V1.1 to SAML V2.0. We assume that your SP software has the ability to issue SAML V2.0 requests and consume SAML V2.0 assertions.

Generally speaking, **before** making any changes to the software configuration, an SP's metadata is updated for SAML V2.0 and allowed to propagate throughout the Federation. Since Web Browser SSO almost always begins at the SP, exposing endpoints in SP metadata that are not supported in software is usually harmless. On the other hand, issuing SAML V2.0 requests without appropriate SAML V2.0 endpoints in metadata is a recipe for disaster!

Configuring the SP

This section shows how to update metadata and configure the SP software for SAML V2.0 Web Browser SSO.

Preconditions:

- The organization responsible for the SP is an InCommon Federation participant
- The SP software supports SAML V2.0 Web Browser SSO
- A deployment choice with respect to IdP discovery (e.g., the SAML V2.0 Identity Provider Discovery Protocol) has been made

Procedure:

1. Update InCommon metadata for SAML V2.0
    - Add one or more SAML V2.0 endpoints to metadata
    - Add an encryption key to metadata
2. Wait for the newly updated metadata to propagate throughout the Federation
3. Configure the software for SAML V2.0 Web Browser SSO
    - Configure the software with the corresponding decryption key
    - Configure the software for IdP discovery
    - Configure the software to issue SAML V2.0 authentication requests
    - Configure the software to consume SAML V2.0 assertion responses

Procedural details:

InCommon metadata is updated at step 1 in advance of configuring the software for SAML V2.0. First add one or more SAML V2.0 endpoints to metadata, including at least one `<md:AssertionConsumerService>` endpoint and zero or more `<idpdisc:DiscoveryResponse>` endpoints.

Since a SAML V2.0 IdP typically encrypts assertions transmitted through the browser, the SP is obliged to add an encryption key to metadata as well. In the InCommon Federation, this will already be done since a key in SP metadata is designated as a multiple use key by default. (See the wiki topic on Key Usage for details.)

At step 2, you must wait for the new metadata to propagate before continuing with the remaining steps. We recommend you **wait at least three (3) days for the metadata to propagate**, but you may have to wait longer if your partners do not routinely refresh metadata.

At step 3, begin by configuring the software with the private decryption key corresponding to the public encryption key in metadata. If an encryption key was already in metadata when you started this procedure, perhaps the decryption key is likewise already configured in software. Double-check your configuration to be sure.

If the SP deployment will use the SAML V2.0 Identity Provider Discovery Protocol, the software is configured to issue such protocol requests in the presence of an unauthenticated user. Otherwise this configuration step may be omitted in favor of some other approach to IdP discovery.

Finally the software is configured to issue SAML V2.0 authentication requests and consume SAML V2.0 assertion responses. One or more endpoint configurations are required, depending on the `<md:AssertionConsumerService>` endpoint(s) added to metadata at step 1.

Testing the SP

Once the SP has been upgraded to SAML V2.0, a natural tendency is test the complete, end-to-end flow. If this works, you may be done, but if it doesn't, or you require more thorough testing, a targeted test sequence may be employed:

1. Test the SP's ability to consume a SAML V2.0 assertion response
2. Test the SP's ability to issue a SAML V2.0 authentication request
3. Test the SP's ability to issue a SAML V2.0 Identity Provider Discovery Protocol request

Such tests limit the scope of the problem and therefore make bugs easier to find.