Identity Provider for SAP Single Sign-On and SAP Identity Management
# Content

1. **Identity Provider for SAP Single Sign-On and SAP Identity Management.**

   1.1 What is SAML 2.0. ........................................... 5
   
      SSO with SAML 2.0. ............................................ 6
      SLO with SAML 2.0. ............................................. 9
      Identity Federation. ............................................. 10
      Common Domain and Identity Provider Discovery. ........ 14
      Identity Provider Proxy. ....................................... 16

   1.2 Before Starting. .............................................. 23
   
      System Requirements. .......................................... 23
      Authorizations. .................................................. 24
      Limitations of the Identity Provider. ......................... 25

   1.3 Adding an Identity Provider to Your Network. ............... 26
   
      Downloading and Installing the Federation Software. .... 26
      Configuring the Identity Provider. .......................... 27
      Enabling the SAML Identity Provider. ....................... 29
      Configuring Back-Channel Communication. .................. 31
      Configuring Front-Channel Communication. .................. 35
      Configuring Support for Enhanced Client or Proxy. ....... 38
      Trusting Service Providers. ................................... 39

   1.4 Optional Configurations. ..................................... 52
   
      Adding Custom Authentication Contexts. ..................... 52
      Configuring the Identity Provider for Discovery With CDCs. 53
      Configuring Identity Providers as Proxies. .................. 55
      Configuring Identity Provider Extensions. .................. 56
      Configuring Policy Scripts for Identity Provider Extensions. . 65
      Configuring the Metadata and Metadata Access. ............ 69
      Configuring the Redirect Application. ....................... 70
      Configuring the Lifetime of Identity Provider Sessions. .... 71
      Configuring the Validity Period for SAML Messages. ....... 72
      Disabling IdP-Initiated and SP-Initiated SSO and SLO. .... 73
      Disabling the SAML 2.0 Provider. ............................. 73
      Enabling HTTP Access to SAML Endpoints. .................. 74
      Enabling Service Providers to Share Persistent Name IDs. ... 75
      Determining the Channel Used for SLO by the Identity Provider. 76
      Including Legacy Systems in Your SAML 2.0 Landscape. ...... 76
      Mapping Authentication Contexts to Log-in Modules. ......... 77
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Principles and Configuration of the Scoping Element.</td>
<td>79</td>
</tr>
<tr>
<td>Securing SAML Bindings.</td>
<td>80</td>
</tr>
<tr>
<td>Setting the Timeout for Database Lock in Clusters.</td>
<td>82</td>
</tr>
<tr>
<td>1.5 Operations and Monitoring.</td>
<td>83</td>
</tr>
<tr>
<td>Managing Name IDs.</td>
<td>83</td>
</tr>
<tr>
<td>Monitoring Identity Provider Sessions.</td>
<td>84</td>
</tr>
<tr>
<td>Auditing and Performance Monitoring.</td>
<td>85</td>
</tr>
<tr>
<td>Troubleshooting on the Identity Provider.</td>
<td>86</td>
</tr>
<tr>
<td>Performing Identity Provider-Initiated Single Sign-On.</td>
<td>86</td>
</tr>
<tr>
<td>Performing Identity Provider-Initiated Single Log-Out.</td>
<td>89</td>
</tr>
<tr>
<td>Accessing the Metadata XML of SAML Identity Providers.</td>
<td>90</td>
</tr>
</tbody>
</table>
1 Identity Provider for SAP Single Sign-On and SAP Identity Management

Document History

The following table provides an overview of the most important document changes.

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>6/9/2010</td>
<td>Initial release.</td>
</tr>
<tr>
<td>1.05</td>
<td>12/6/2010</td>
<td>Added optional configuration for adding authentication contexts and mapping to login modules. Added configuration of metadata and metadata access. Moved conceptual description of identity provider proxy to first chapter. Added configuration deletion function. Updated system requirements. Updated description of SCA download.</td>
</tr>
<tr>
<td>1.10</td>
<td>7/18/2011</td>
<td>Updated the description of using common domain cookie for identity provider discovery. Updated system requirements. Updated description of SCA download.</td>
</tr>
</tbody>
</table>
| 1.15    | 7/23/2012 | • Updated the scenarios of using identity provider proxy, including the configuration of the redirect application and the setting of the scoping element. For more information, see Identity Provider Proxy [page 16], Configuring the Redirect Application [page 70], and Principles and Configuration of the Scoping Element [page 79].  
  • Updated the description of using attributes in out-of-band account linking, in identity federation with persistent pseudonyms, and in identity federation with transient users. For more information, see Trusting Service Providers [page 39]. |
| 2.0     | 11/12/2012| Updated the identity provider proxy scenario with the features of the new field Default Proxy Target. For more information, see Identity Provider Proxy [page 16]. |
1.1 What is SAML 2.0

The Security Assertion Markup Language (SAML) version 2.0 is a standard for the communication of assertions about principals, typically users. The assertion can include the means by which a subject was authenticated, attributes associated with the subject, and an authorization decision for a given resource.

The main benefits of SAML 2.0 are as follows:

- **SSO with SAML 2.0**
  SAML provides a standard for cross-domain Single Sign-On (SSO). Other methods exist for enabling cross-domain SSO, but they require proprietary solutions to pass authentication information across domains. SAML 2.0 supports identity-provider-initiated SSO as in SAML 1.x. SAML 2.0 also supports service-provider-initiated SSO.

- **SLO with SAML 2.0**
  Single Log-Out (SLO) enables users to cleanly close all their sessions in a SAML landscape, even across domains. Not only does this save system resources that would otherwise remain reserved until the sessions time out, but SLO also mitigates the risk of the hijacking of unattended sessions.

- **Identity federation**
  Identity federation provides the means to share identity information between partners. To share information about a user, partners must be able to identify the user, even though they may use different identifiers for the same user. The SAML 2.0 standard defines the name identifier (name ID) as the means to establish a common identifier. Once the name ID has been established, the user is said to have a federated identity.

The two main components of a SAML 2.0 landscape are an identity provider and a service provider. The service provider is a system entity that provide a set of Web applications with a common session management, identity management, and trust management. The identity provider is a system entity that manages identity...
information for principals and provides authentication services to other trusted service providers. In other words, the service providers outsource the job of authenticating the user to the identity provider. The identity provider maintains the list of service providers where the user is logged in and passes on log-out requests to those service providers. The client that is trying to access the resource must be HTTP-compliant.

### 1.1.1 SSO with SAML 2.0

SAML provides a standard for cross-domain Single Sign-On (SSO). Other methods exist for enabling cross-domain SSO, but they require proprietary solutions to pass authentication information across domains. SAML 2.0 supports identity provider-initiated SSO as in SAML 1.x. SAML 2.0 also supports service provider-initiated SSO.

When the identity provider initiates SSO, you must maintain links on the identity provider system to the protected resources on the service providers. When you protect resources with SAML on a service provider, the service provider is configured to request authentication from the identity provider. You can also allow the system to redirect the user to a specified redirect site.

#### Features

SAML provides options to pass SAML messages back and forth between the identity provider and the service provider.

- **Front channel**
  - SAML messages are passed back and forth through the user agent with HTTP redirect or HTTP POST methods.

- **Back channel**
  - Only SAML artifacts are exchanged through the identity provider and service provider through the user agent. When a provider receives an artifact, it queries the other provider directly over SOAP.

Back-channel communication provides additional security, by ensuring that potential eavesdroppers of the user agent only access the SAML artifacts. However, back-channel communication requires additional round trips to resolve an authentication request. You can protect front-channel communication with encryption and digital signatures. You can mix the communication options.

#### Front-Channel Communication

The following figure illustrates service-provider-initiated SSO with front-channel communication.
1. A user attempts to access a resource protected by SAML 2.0.
2. The service provider redirects the user to an identity provider for authentication.
3. The identity provider queries the user for authentication credentials.
4. The user or user agent presents the requested credentials.
5. The identity provider returns the user to the service provider with an authentication response.
6. The service provider presents the requested resource to the user.

**Back-Channel Communication**

The following figure illustrates service-provider-initiated SSO with back-channel communication.
1. A user attempts to access a resource protected by SAML 2.0.

2. The service provider redirects the user to an identity provider and includes a SAML artifact referring to the authentication request.

3. The identity provider retrieves the authentication request from the service provider over a SOAP channel.

4. The identity provider queries the user for authentication credentials.

5. The user or user agent presents the requested credentials.

6. The identity provider returns the user to the service provider with a SAML artifact referring to the authentication response.

7. The service provider retrieves the authentication response from the identity provider over a SOAP channel.

8. The service provider presents the requested resource to the user.

**Related Information**

Configuring the Redirect Application [page 70]
### 1.1.2 SLO with SAML 2.0

Single Log-Out (SLO) enables users to cleanly close all their sessions in a SAML landscape, even across domains. Not only does this save system resources that would otherwise remain reserved until the sessions time out, and SLO also mitigates the risk of the hijacking of unattended sessions.

#### Features

SAML provides a number of binding options to pass SAML messages back and forth between the identity provider and the service provider.

- **Front channel**
  - For front-channel communication, SAML messages are passed back and forth over the user agent with HTTP redirect or HTTP POST methods.

- **Back channel**
  - For back-channel communication, the identity provider and service provider can use either SAML artifacts or communicate directly over SOAP. For SAML artifacts, the identity provider and service provider exchange SAML artifacts over the user agent. When a provider receives an artifact, it queries the other provider directly over SOAP to resolve the artifact. For the SOAP binding, the providers pass no artifacts. They exchange SAML messages directly over SOAP.

Back-channel communication provides additional security, by ensuring that potential eavesdroppers of the user agent cannot access the SAML messages. However, the artifact binding requires additional round trips to resolve an authentication request. You can protect front-channel communication with encryption and digital signatures. You can mix the communication options.

The figure below illustrates SLO initiated at the service provider over a front-channel binding, such as HTTP redirect, and between the identity provider and the other service providers over a back-channel binding, such as SOAP over HTTP.
1. The user initiates a log-out request at a service provider.
2. The service provider forwards this request to an identity provider.
3. After the identity provider validates the request, it sends new log-out requests to all other service providers, with which the user has a security session that the identity provider is aware of.
4. The service providers validate the request, destroy any session information for the user, and send a log-out response to the identity provider.
5. The identity provider destroys the user’s sessions and sends a response to the original service provider.
6. The original service provider informs the user that he or she has been logged out.

1.1.3 Identity Federation

Identity federation provides the means to share identity information between partners. To share information about a user, partners must be able to identify the user, even though they may use different identifiers for the same user. The SAML 2.0 standard defines the name identifier (name ID) as the means to establish a common identifier. Once the name ID has been established, the user is said to have a federated identity.

The SAML 2.0 standard defines a number of name ID formats. The table below describes the name ID formats.

<table>
<thead>
<tr>
<th>Name ID Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail</td>
<td>The name ID is an e-mail address.</td>
</tr>
<tr>
<td>Kerberos</td>
<td>The name ID is a Kerberos Principal Name (KPN).</td>
</tr>
</tbody>
</table>
### Name ID Format

<table>
<thead>
<tr>
<th>Name ID Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Persistent</strong></td>
<td>The name ID is a permanent opaque string generated by the identity provider for a service provider or an affiliation of service providers.</td>
</tr>
<tr>
<td><strong>Transient</strong></td>
<td>The name ID is a temporary opaque string generated by the identity provider for a service provider for the lifetime of a security session.</td>
</tr>
<tr>
<td><strong>Unspecified</strong></td>
<td>The implementation of this name ID is vendor-specific. SAP assumes this value to be either a user ID, a logon alias, or another attribute value of the user.</td>
</tr>
<tr>
<td><strong>Windows Name</strong></td>
<td>The name ID is a user ID qualified by a Microsoft Windows domain.</td>
</tr>
<tr>
<td><strong>X509 Subject Name</strong></td>
<td>The name ID is the subject name of an X.509 certificate.</td>
</tr>
</tbody>
</table>

### Types of Federation

SAML describes the following types of federation:

- Out-of-band account linking
- Transient pseudonym identifiers
- Persistent pseudonym identifiers

### Out-of-Band Account Linking

The identities of a user in system A and system B are identified and agreed upon ahead of time between the administrators of the two systems. This kind of agreement is also supported by SAML 1.x. The administrator of the identity provider and the service provider agree how the name ID used for the user in the identity provider maps to the user in the service provider.

#### Example

Users in the identity provider always log on with their e-mail address. The log-on ID and e-mail address are identical. The administrator of the identity provider agrees to provide the Unspecified name ID format including the log-on ID. After a user successfully logs on to the identity provider, whether by Kerberos name or client certificate or some other means, the identity provider provides the log-on ID of the user to the service provider in the SAML assertion. The service provider is also configured to use the Unspecified name ID format and is configured to use the user attribute for the e-mail address. The service provider searches for the user with an e-mail address that matches. As long as the e-mail address in the service provider is unique, the service provider can log the user on.

The figure below shows Laurent Becker has different user IDs on the identity provider and service provider. With SAML 2.0, he authenticates on the identity provider. The identity provider passes his user ID to the service provider, and the service provider searches for his user using his e-mail address. Thus his two accounts are linked by user ID and e-mail address.
Account Linking with E-Mail Address

Use this kind of federation to support most scenarios where you need to map user identities across domains.

**Transient Pseudonym Identifiers**

Federation with transient name IDs creates a federation with a temporary user in the service provider and a permanent user in the identity provider. This federation only exists as long as the security session with the service provider exists. The service provider does not persist data about the visiting user. User attributes and access rights are generated based on rules applied to attributes sent in SAML messages.

Use this kind of federation when the service provider does not need to record information about users or does not need local user accounts.

**Persistent Pseudonym Identifiers**

Federation with persistent name IDs establishes a permanent relationship between a user on an identity provider and a user on a service provider or users on an affiliation of service providers. The persistent name ID is used by an identity provider and a service provider as a common name for a single user. If this name ID for a user is the same for multiple service providers, the service providers are said to be affiliated or belong to an affiliation group.

Use this kind of federation to link accounts out-of-band, but without using identifiers that can be traced back to a specific user. This increases the security of your systems by preventing eavesdroppers from determining...
identities on the basis of name ID formats that pass log-on IDs or e-mail addresses. It requires you to establish the pseudonym on both providers ahead of time.

Federation with persistent name IDs also offers the following additional options:

- **Interactive federation**
  Federation is established on the fly. You can enable users to interactively establish federation between existing accounts or even create their own account on the target system with self-registration. Use this kind of federation if you have not created persistent pseudonyms on the identity provider and service provider ahead of time. It enables you to configure these mappings as you go.

- **Automatic creation**
  Federation is established on the basis of attributes passed to the target system. If the user has no account in the target system, the service provider automatically creates the account. The attributes are generated from rules based on SAML 2.0 attributes sent in SAML messages. Use this kind of federation to create and even provision users as you federate their accounts on the service provider.

**Example**

The figure below shows that the accounts of Laurent Becker each have an attribute for a persistent name ID, named opaque ID. The value to use here can be agreed upon in advance by the two system administrators or generated by the identity provider and distributed to the service provider. When Laurent Becker authenticates on the identity provider, the service provider receives the SAML assertion with the opaque ID as the subject. The service provider searches for the user based on the opaque ID and logs the user on.
Securing Data

You can ensure that sensitive data is encrypted when two systems share information of this type. The use of the pseudonym name ID formats (transient and persistent) ensure privacy and anonymity between two partners (identity provider and service provider). Neither partner needs to be aware of the local account name used by the other partner, protecting the user’s privacy.

1.1.4 Common Domain and Identity Provider Discovery

Security Assertion Markup Language (SAML) 2.0 service providers can use a common domain cookie (CDC) to determine to which identity provider they should send a request. The common domain is the domain where the CDC resides. This common domain is known to both the identity provider and the service provider. Identity providers identify themselves to service providers by writing their alias into the CDC. The service provider reads the alias from the CDC. This identity provider includes an internal write service to write its alias into the CDC. It can also use an external write service. When enabled, these services write CDCs to help the identity provider to identify itself to service providers.

When to use the external and internal write services depends on your network architecture.

- If the identity provider shares the same domain with the common domain, use the internal service.
- If the identity provider exists in a different domain from the common domain, use the external service.

Common Domain is the Shared Domain

The figure below illustrates a service provider and an identity provider sharing the same domain. The identity provider writes its alias to a CDC in the shared domain using domain relaxing to remove its host name. The internal read service of the service provider can read the CDC because it shares the same domain.
Common Domain is a Different Domain

The figure below illustrates a service provider and an identity provider in two different domains. The operators of both providers have agreed on a common domain for the CDC at itelo.biz. The identity provider calls an external write service to write its alias to the CDC in the common domain. The service provider calls an external read service within the common domain to read the CDC. The external read service of the service provider can read the CDC because the read service shares the same domain with the CDC.
Service Provider and Identity Provider Reside in Different Domains and Access Common Domain Cookie in Common Domain

Related Information

Configuring the Identity Provider for Discovery With CDCs [page 53]

1.1.5 Identity Provider Proxy

An identity provider can function as a proxy for another identity provider. An identity provider proxy enables you to create structures of trust relationships that ultimately simplify the management of your service providers.

A proxy relationship involves the following participants:

- Authenticating Identity Provider
  The authenticating identity provider trusts the service provider of the identity provider proxy.
- Identity Provider Proxy
  The identity provider proxy is an identity provider and a service provider. The service provider of the identity provider proxy trusts the authenticating identity provider.
- Target System
  A service provider hosts a service that users want to access. This service provider trusts the identity provider of the identity provider proxy.
There is no direct trust relationship between the authenticating identity provider and the service provider that the user is trying to access. The following figure illustrates this relationship.

**Trust Relationships of an Identity Provider Proxy**

**Propagation of Authentication Requests and Responses in a Proxy Configuration**

SAML 2.0 uses a proxy count to limit how far up and down a chain of providers authentication requests and responses can go. A provider can include a proxy count in the request or response. In the SAP solution, if the message does not include a proxy count, the identity provider inserts the proxy count restriction from the identity provider settings for authentication responses and from the service provider settings for authentication requests. Every provider the message passes through reduces the count by 1 until either the
message is resolved or the count reaches 0. A provider that receives a message with a proxy count of 0 rejects the message. To configure the proxy count restriction for the identity provider, choose the Proxying Settings tab and enter the value in the Proxy Count (Assertion) field. You also have to make sure that you have selected an operation mode for both the identity provider and the service provider. To configure the operation mode, choose the tabs Authentication and Single Sign-On SAML 2.0 Local Provider.

In the SAP solution, when an identity provider receives an authentication request, assuming the proxy count has not reached 0, it checks if proxying is enforced. You can configure the enforce proxy settings by choosing the Proxying Settings tab.

- **Proxying Is Enforced**
  - If the authentication request from the service provider to the identity provider proxy does not specify to what identity provider the service provider must be proxied, the identity provider proxy will use the identity provider specified in the Default Proxy Target field for such proxying. If no such identity provider is specified, the proxying in that case will fail.

- **Proxying Is Not Enforced**
  - If the authentication request does not include a list of authenticating identity providers, the identity provider proxy tries to authenticate the user.
  - If the authentication request includes a list of authenticating identity providers, and the identity provider proxy is in this list, the identity provider proxy tries to authenticate the user.
  - If the authentication request includes a list of authenticating identity providers, but none of them is trusted, the system directs the user to the default identity provider trusted by the identity provider proxy.
  - If the authentication request includes a list of authenticating identity providers and at least one of them is trusted by the identity provider proxy, the system directs the user to the first trusted authenticating identity provider.

Other scenarios regardless of the enforce proxy settings:

- If the authentication request does not include a list of authenticating identity providers or none of them is trusted, the system directs the user to the default identity provider trusted by the identity provider proxy.
- If the authentication request includes a list of authenticating identity providers and at least one of them is trusted by the identity provider proxy, the system directs the user to the first trusted authenticating identity provider.

### Principles and Configuration of the Scoping Element

You use the scoping element for the identity provider proxy scenarios. By setting the option to send the scoping element, you specify a list of identity providers that is included in the authentication request. However, it is also possible to use authenticating identity providers that do not support the scoping element.

**Note**

The option Send Scoping Element is set to Yes by default.
Principles and Configuration of the Redirect Application

You use the redirect application to allow the system to redirect you to another site. The system redirects the user to the specified redirect site by a GET method (the redirect site application does not preserve the POST method). In addition, you can set parameters to the redirect URL address. You can configure the redirect site mappings by choosing Configuration Management > Authentication and Single Sign-On > SAML 2.0 > Local Provider > Proxying Settings.

Service Provider Initiated Single Sign-On (SSO) and Single Log-Out (SLO)

In the SSO scenario the service provider sends an authentication request to the identity provider proxy, and the identity provider proxy decides whether to authenticate the user. The identity provider proxy makes such a decision in accordance with the enforced proxying settings or from the information in the authentication request. This means if the proxying is not enforced, the authentication request signifies whether or not the identity provider proxy will authenticate the user. However, if the SAML 2.0 authentication fails at the identity provider proxy or further down in the chain, the user will be allowed to authenticate with user name and password. If the authentication with user credentials succeeds, the SSO scenario proceeds with the identity provider issuing an assertion based on the logged-in user. In a case of this type, the system uses the BasicPasswordLoginModule, which is configured as mapped from the PasswordProtectedTransport authentication context in HTTPS access or from the Password authentication context in HTTP access. This means that the user can log in locally when the SAML 2.0 authentication fails at the identity provider proxy. In a similar way, the SLO procedure triggered at the service provider of the target system results in a log-out request sent to the identity provider proxy. Consequently, the identity provider proxy processes the request and destroys any local session information about the user. The identity provider proxy then checks whether there are other service providers in the SSO session and sends log-out requests to all of them. In return, the service providers send log-out responses to the identity provider proxy informing it that the log-out process is successful. Finally, the identity provider proxy sends a log-out response to the original requesting service provider or the service provider of the target system, and this procedure completes the log-out process. In this scenario the authenticating identity provider does not receive a log-out request and is not informed of the log-out process.

Identity Provider Initiated Single Sign-On (SSO) and Single Log-Out (SLO)

To initiate SSO from an authenticating identity provider, the user should have established a valid security context with the authenticating identity provider. Otherwise, the user will be challenged to provide credentials in order to establish a security context. Once the user has a valid security context, the authenticating identity provider issues an SAML 2.0 assertion and sends it to the identity provider proxy. The identity provider proxy validates the assertion, and if the validation is successful, it creates a security context for the user itself. Consequently, the identity provider proxy issues a new SAML 2.0 assertion based on the newly created security context for the target system to complete the SSO process. Finally, the service provider at the target system creates its own local security context.
Identity Provider Initiated SSO

When the user initiates the SLO process at the authenticating identity provider, the authenticating identity provider sends a log out request to the identity provider proxy directly or through the browser, depending on the type of communication. The following types of communication exist here:

- Front-Channel SLO
  In the front-channel scenario, the identity provider proxy sends a request through the browser to the service provider. The service provider returns a log out response to the identity provider proxy by using the redirect or post binding. The identity provider proxy also sends a log-out response to the authenticating identity provider. Finally, the authenticating identity provider informs the user that the log out request is successful.

- Back-Channel SLO
  In the back-channel scenario the identity provider proxy sends the log-out request to the service provider directly. The service provider returns a log-out response to the identity provider proxy using the SOAP binding. With this direct system-to-system communication, the identity provider proxy sends a response to the authenticating identity provider, which informs the user of the result in the log-out process.

Related Information

- Principles and Configuration of the Scoping Element [page 79]
- Configuring the Redirect Application [page 70]
1.1.5.1 Examples

Identity Provider initiated SSO

In the identity provider initiated SSO, the identity provider of the identity provider proxy is configured with specialized links that refer to the target service providers. These links actually refer to the local identity provider’s SSO service and pass parameters to the service identifying the remote service provider.

The user John is willing to initiate a SSO process at the identity provider, and he has configured the link to the target service provider. He tries to access the authenticating identity provider, but because he does not have a valid session, he is asked to provide credentials. After John logs in, a log-on security context is created for him at the authenticating identity provider, and he is directed to the identity provider proxy. John chooses the link to the service provider, and the system directs him to the web site of the service provider. After the assertion consumer service of the service provider validates the assertion passed from the identity provider proxy, the service provider creates a local security context for John and gives John access rights.

SAML 2.0 to SAP Log-on Ticket Scenario of SSO

A company hosts applications for three customers, companies A, B, and C. These applications run on an old ABAP system that does not support SAML 2.0. Each customer uses a separate ABAP client so that there is isolation between companies’ data. The users from the companies A, B, and C have accounts in the ABAP system and can log on with user name and password. However, the hosting company would like to make it easier for the users by introducing SSO to the ABAP system. This procedure would prevent users from having to reenter their user names and passwords to access each separate application.

Users from company A access an application on the ABAP system and, as usual, they see the log-on screen. In addition, they also see a link that points to the proxy application – for example “Single Sign-On Authentication”. If the users choose the “Single Sign-On Authentication” link, they are directed to the SAML 2.0 proxy application hosted on the service provider system of the hosting company. Since they do not yet have a session, they are directed to the identity provider of company A for authentication. After authentication at the identity provider, the identity provider sends a SAML 2.0 response to the proxy application, which evaluates the response, authenticates the users, creates an SAP log-on ticket, and directs the log-on ticket back to the accessed application on the ABAP system. The ABAP system evaluates the SAP log-on ticket and authenticates the users. As a result, the users have access to the application.

Sharing Trust Between Organizations

Company B provides services to Company A. The administrator of Company B uses an identity provider proxy to manage trust between his service providers and the identity provider of Company A. When employees of Company A access Company B’s service providers, the service providers send the authentication request to the identity provider proxy and then on to the authenticating identity provider at Company A. When employees of Company B access Company B’s service providers, the service providers send the authentication request to the identity provider proxy where the request is resolved. You can use the URL parameter saml2idp to determine which identity provider should authenticate the authentication request. With this configuration, administrators can ensure that their users are authenticated by their own identity provider. When the
The administrator of Company B adds a new service provider, he only needs to configure trust between his local service provider and the identity provider proxy. He does not need to contact the administrator of Company A and ask her to configure trust in her identity provider.

The figure below illustrates the scenario.

---

**Load Balancing**

You can achieve a measure of load balancing by creating a pyramid of identity providers and service providers. At the top of the pyramid, you have a single authenticating identity provider. Below this you have a number of identity provider proxies. At the bottom you have service providers trusting identity provider proxies. The identity provider proxies should never authenticate users. You can ensure this by configuring the Enforce proxying option under the Proxying Settings tab. Instead they pass the request on to the authenticating identity provider. The authenticating identity provider establishes an identity provider session and passes the request back to the identity provider proxy. The identity provider proxy establishes a session in turn. The next time the user accesses another service provider within the same group, the session is already established on the identity provider proxy. Only when the user accesses a service provider in a different group does the authentication request have to go back up to the authenticating identity provider.

The figure below illustrates the scenario.
1.2 Before Starting

Review the following before you install and configure the identity provider for SAP NetWeaver Application Server (AS) Java.

1.2.1 System Requirements

- To support the identity provider extensions, the host SAP NetWeaver Application Server (AS) Java must be of the following releases:
  - AS Java 7.3 SPS 13 or higher
  - AS Java 7.31 SPS 15 or higher
  - AS Java 7.4 SPS 10 or higher
  - AS Java 7.5 - any support package stack
- To support the newest user interface improvements, the host SAP NetWeaver Application Server (AS) Java must be of release AS Java 7.2 SPS 04 or higher.
User interface improvements include functions to add authentication contexts and map them to log-in modules, to configure metadata and metadata access, and to delete the identity provider configuration. Otherwise the host AS Java must be of the following releases:

- AS Java 7.2 SPS 02 with 1471322 applied
- AS Java 7.2 SPS 03 or higher

- You must have SAP Single Sign-On (SAP SSO) 2.0 or higher, or SAP Identity Management 7.2 or higher, installed in your system landscape.

For more information about licensing SAP products, consult your key account manager.

1.2.2 Authorizations

To work with the administration and configuration user interfaces of the identity provider, you must log on with a user that has the required authorizations. The following table lists the user management engine (UME) actions required for the identity provider user interfaces.

UME Actions for the Identity Provider User Interfaces

<table>
<thead>
<tr>
<th>Service/Application</th>
<th>Name</th>
<th>Description</th>
<th>Default Role Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>saml2_cfg</td>
<td>editSAML2Cfg</td>
<td>Provides read-write access to the SAML 2.0 and Key Storage Web Dynpro applications.</td>
<td>Administrator, NWA_SUPERADMIN, SAML2_SUPERADMIN</td>
</tr>
<tr>
<td>saml2_cfg</td>
<td>viewSAML2Cfg</td>
<td>Provides read-only access to the SAML 2.0 and Key Storage Web Dynpro applications.</td>
<td>NWA_READONLY, SAML2_READONLY</td>
</tr>
<tr>
<td>saml2_idp_admin</td>
<td>modifySAML2IdPAdmin</td>
<td>Provides read-write access to the Identity Provider Sessions Web Dynpro application.</td>
<td>Administrator, SAML2_SUPERADMIN</td>
</tr>
<tr>
<td>saml2_idp_admin</td>
<td>viewSAML2IdPAdmin</td>
<td>Provides read-only access to the Identity Provider Sessions Web Dynpro application.</td>
<td>SAML2_READONLY</td>
</tr>
</tbody>
</table>

The following UME actions are reserved for future use:

- SAML2UserLockedAction and the corresponding role SAML2UserLocked
- BackchannelEndpoint

The SAML 2.0 implementation includes the roles listed in the table below with the AS Java.

UME Roles for SAML 2.0

<table>
<thead>
<tr>
<th>Name</th>
<th>Assigned Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML2_READONLY</td>
<td>- viewSAML2IdPAdmin</td>
<td>Provides read-only access to the Identity Provider Sessions, SAML 2.0, and Key Storage Web Dynpro applications.</td>
</tr>
<tr>
<td></td>
<td>- viewSAML2Cfg</td>
<td></td>
</tr>
<tr>
<td>SAML2_SUPERADMIN</td>
<td>- editSAML2Cfg</td>
<td>Provides read-write access to the Identity Provider Sessions, SAML 2.0 and Key Storage Web Dynpro applications.</td>
</tr>
<tr>
<td></td>
<td>- modifySAML2IdPAdmin</td>
<td></td>
</tr>
</tbody>
</table>
i Note

Do not consider the access, which these roles and actions grant to the Key Storage application, sufficient for general usage of that application, but enough for the administration of your SAML 2.0 configuration.

These roles and the actions for viewing and editing the SAML 2.0 configuration application are part of the AS Java installation. When you install the identity provider, the installer adds the actions for the identity provider session application. If these roles were deleted before you install the identity provider, the installer re-creates the roles and includes only the actions for the identity provider session application.

1.2.3 Limitations of the Identity Provider

SAP Identity Management (IDM) supports the IDP implementation of the Security Assertion Markup Language (SAML) version 2.0. The following section describes implementation consideration for the use of SAP Identity Management as a SAML 2.0 provider.

Transient Pseudonyms and Auditing

SAP Identity Management supports auditing of transient pseudonym federation by recording for what user a transient name ID was create and when in the security audit log. However, the service provider must also support auditing to identify the real user behind the transient name ID.

Identity Provider-Initiated Single Log-Out and Microsoft Internet Explorer

→ Recommendation

To avoid problems during Single Log-Out (SLO) initiated from the identity provider when using Microsoft Internet Explorer 6 or 7, we recommend that you configure the browser to check for new versions of stored pages on every visit.
1.3 Adding an Identity Provider to Your Network

This section includes an outline of the procedures required to download and configure the identity provider for SAP NetWeaver Application Server (AS) Java.

1.3.1 Downloading and Installing the Federation Software

Context

As of SAP Single Sign-On 2.0 and SAP Identity Management 7.2, the federation software component archive (SCA) includes the identity provider and the security token service software.

As of SAP Identity Management 8.0 SP04, the federation software is delivered as *.SAR file.

Depending on your SAP Identity Management version, there are different procedures for downloading the federation software.

SAP Identity Management 7.2

Procedure

2. In the navigation pane, choose SAP Software Download Center > Support Packages and Patches.
3. In the A-Z Index, navigate to the S section.
4. Navigate to the following product: SAP SINGLE SIGN ON > SAP SINGLE SIGN ON 3.0 > Comprised Software Component Versions > IDM FEDERATION 7.2.
5. Download IDM_FEDERATION<Version>.SCA.
6. Deploy the SCA to the AS Java.
   You can use the Deployment Job view of the SAP NetWeaver Developer Studio.

Related Information

SAP Single Sign-On
SAP Identity Management 8.0

Prerequisites

If you are running SAP Identity Management 8.0 SP04 or higher, you require the SAPCAR archiving tool to be able to unpack software component archives (*.SAR files).

Procedure

2. In the navigation pane, choose SAP Software Download Center > Software Downloads > Support Packages and Patches.
3. In the A-Z Index, navigate to the N section.
4. Navigate to the following product: SAP NW IDENTITY MANAGEMENT > SAP IDENTITY MANAGEMENT 8.0 > Comprised Software Component Versions > IDM FEDERATION 7.2.
5. Depending on your SAP Identity Management 8.0 SP release, download the following:
   - For version 8.0 SP03 or lower: IDM FEDERATION<Version>.SCA
   - For version 8.0 SP04 or higher: IDM FEDERATION<Version>.SAR
6. For version 8.0 SP04 or higher, unpack the installation kit using the following command:
   ```bash
   SAPCAR -xvf <your-SAR-file>
   ```
7. Deploy the IDM FEDERATION<Version>.SCA to the AS Java.
   You can use the Deployment Job view of the SAP NetWeaver Developer Studio.

1.3.2 Configuring the Identity Provider

This procedure provides an overview of the steps to configure SAP NetWeaver Application Server (AS) Java as a Security Assertion Markup Language (SAML) 2.0 identity provider. As an identity provider, the AS Java enables you to off-load the authentication of users from service providers. The identity provider enables you to federate identities across domains for Single Sign-On (SSO). Once logged on, SAML 2.0 enables Single Log-Out (SLO).

Prerequisites

- You have created any necessary keys and certificates in a keystore view dedicated to SAML. For more information, see the documentation for the keystore manager of the AS Java.
- There is a SAML 2.0 service provider in your SAML network.
  The service provider can be in the same local area network or in another domain.
**Procedure**

1. Enable SAML 2.0 support and select the certificates for digital signatures and encryption.
   For more information, see *Enabling the SAML Identity Provider*.

2. Determine how your identity provider communicates with service providers.
   For more information, see the following:
   - Configuring Front-Channel Communication
   - Configuring Back-Channel Communication
   - Configuring Support for Enhanced Client or Proxy

3. Trust a service provider.
   For more information, see *Adding Service Providers*.

4. Determine how to federate the identities on the identity provider and service provider.
   For more information, see the following:
   - Configuring Out-of-Band Account Linking
   - Configuring Identity Federation with Transient Users
   - Configuring Identity Federation with Persistent Pseudonyms

5. Make any optional configurations.
   For more information, see *Optional Configurations*.

**Related Information**

- Enabling the SAML Identity Provider [page 29]
- Configuring Front-Channel Communication [page 35]
- Configuring Back-Channel Communication [page 31]
- Configuring Support for Enhanced Client or Proxy [page 38]
- Adding Service Providers [page 39]
- Configuring Out-of-Band Account Linking [page 44]
- Configuring Identity Federation with Transient Users [page 47]
- Configuring Identity Federation with Persistent Pseudonyms [page 49]
- Optional Configurations [page 52]
1.3.3 Enabling the SAML Identity Provider

Use this procedure to enable Security Assertion Markup Language (SAML) 2.0 support and make the basic configurations for a SAML 2.0 identity provider. This procedure only covers the first steps for preparing your SAP NetWeaver Application Server (AS) Java to operate as a SAML identity provider.

Prerequisites

You have downloaded and installed the federation software.

For more information, see *Downloading and Installing the Federation Software*.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose the SAML 2.0 tab.
3. Continue the configuration for the current state of SAML 2.0 configuration on your system. If you have never configured your system for SAML 2.0, the system displays the following message: System not configured to support SAML 2.0.

Related Information

*Downloading and Installing the Federation Software* [page 26]

1.3.3.1 SAML 2.0 Is Already Configured

Context

i Note
If you were viewing the SAML 2.0 configuration application before you installed the federation software, you must navigate away from the application and return before you can complete the configuration. Otherwise you cannot change the operation mode of the provider.
Procedure

1. Choose Edit.
2. Enter an operation mode for the provider.
   - If there are no resources on the host AS Java you want to protect with SAML, enter Identity Provider.
   - If there are resources on the host AS Java you want to protect with SAML, enter Identity Provider and Service Provider.
3. Save your entries.

1.3.3.2 SAML 2.0 is not Configured

Procedure

1. Choose the Enable SAML 2.0 Support pushbutton.
2. Enter a name for the provider.
3. Enter an operation mode for the provider.
   - If there are no resources on the host AS Java you want to protect with SAML, enter Identity Provider.
   - If there are resources on the host AS Java you want to protect with SAML, enter Identity Provider and Service Provider.
4. Configure the settings for signature and encryption.
   a. Select the keystore view and the key pairs for digital signatures and encryption.
      The AS Java creates the SAML2 keystore view and selects this view as the default view as soon as you enable SAML 2.0. For this view, you must generate at least one public-key certificate. You can also use another view, where you have already created key-pair certificates.
   b. Determine if you want to include the public-key certificate in any digital signatures.
      - If you are using a public-key infrastructure for your SAML network or if the trusted providers otherwise require the inclusion of certificates to verify digital signatures, include the certificate.
      - If you are using self-signed certificates, do not include a certificate.
   c. To provide a means for service providers to validate the metadata of the identity provider, sign the configuration metadata of the identity provider.
5. Continue with the provider settings and enter data as desired.
   The SOAP binding for the single log-out (SLO) service and all the types and bindings for the Manage NameID (MNI) Service are disabled by default. Enabling any of these configurations for SLO and MNI without using them slows down the general performance.
   If you have selected the operation mode Identity Provider and Service Provider, you have to go through the service provider settings. For more information about these settings, see the service provider documentation.

Note

This procedure only covers the initial basic configuration for enabling the SAML 2.0 identity provider. Once the identity provider is enabled, you can modify the bindings and types supported by the identity...
provider, trust an identity provider, configure identity federation, and protect resources with SAML. For more information, see Configuring the Identity Provider.

6. Choose the Finish pushbutton.

Related Information

Configuring the Identity Provider [page 27]

1.3.4 Configuring Back-Channel Communication

Back-channel communication uses HTTP artifact bindings or SOAP bindings to communicate between the service provider and the identity provider. Use back-channel communication to ensure that SAML messages are not exposed to the client and any malicious third-parties eavesdropping on the client. Back-channel communication requires a direct connection between a service provider and an identity provider. If there is a firewall between the providers, direct communication may not be possible. Front-channel communication can improve response time for Single Sign-On (SSO) as it requires fewer roundtrips to authenticate a user.

Prerequisites

- You have determined which back-channel bindings you want to support.

<table>
<thead>
<tr>
<th>Binding</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP artifact</td>
<td>This is the only back-channel binding supported by SAML SSO.</td>
<td>Increases the number of roundtrips required to pass a SAML message, increasing response time.</td>
</tr>
<tr>
<td>HTTP artifact binding</td>
<td>HTTP artifact binding sends a reference to a SAML message over the client. The identity provider and the service provider then use SOAP to exchange the SAML message to which the artifact refers.</td>
<td></td>
</tr>
<tr>
<td>SOAP</td>
<td>Providers exchange SAML messages directly.</td>
<td>Firewalls can block SOAP. A domain name system (DNS) may not be able to resolve the destination of the message. Using a SOAP binding slows down the system because of the additional work of the system with the database.</td>
</tr>
<tr>
<td>SOAP binding</td>
<td>SOAP binding sends messages directly between the identity provider and the service provider without involving the client.</td>
<td></td>
</tr>
</tbody>
</table>

- SAML 2.0 has been enabled on your SAP NetWeaver Application Server (AS) Java.
  For more information, see Enabling the SAML Identity Provider.
Procedure

Related Information

Enabling the SAML Identity Provider [page 29]

1.3.4.1 Disabling Back-Channel Communication

Use this procedure to restrict authentication to front-channel communication.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 Local Provider.
3. Choose the Identity Provider Settings tab.
4. Disable the following bindings:
   ○ For the Single Sign-On service, deselect the HTTP artifact checkbox.
   ○ For the Single Log-Out (SLO) service, deselect the HTTP artifact and SOAP checkboxes.

   i Note
   The SOAP binding is disabled by default.

5. On the General Settings tab, under Artifact Resolution Service in the Mode field, select Disabled.
6. Disable HTTP artifact and SOAP bindings from trusted service providers.
   For more information, see the product documentation for your service provider.
1.3.4.2 Enabling Back-Channel Communication with HTTP Artifact

Use this procedure to accept artifacts and configure the other back-channel parameters.

**Procedure**

1. Enabling and Configuring the Artifact Resolution Service
   a. Start SAP NetWeaver Administrator with the quick link `/nwa/auth`.
   b. Choose SAML 2.0 Local Provider.
   c. Choose the General Settings tab.
   d. Under Artifact Resolution Service in the Mode field, select Enabled.
   e. Enter data as required.
      ○ To ensure that synchronization problems between systems do not interfere with the SAML artifact connections, increase the validity period for artifacts accepted.
      ○ Enter the interval for deleting expired artifacts.
        This property determines how often expired, unresolved artifacts are deleted from the database. Resolved artifacts are deleted immediately. You can estimate how quickly artifacts are added to the system based on the number of users you expect to log onto your system at the same time. If you expect heavy usage and space is an issue for your database, set a lower value.

      △ Caution
      If you set a value that is too high, your database fills with expired artifacts causing a decrease in system performance or, in the worst case, causing your system to become unresponsive.

2. Determining Which Services Accept Artifacts
   On the Identity Provider Settings tab, determine the services for which you want to accept artifacts from identity providers.
   ○ To accept artifacts for Single Sign-On (SSO):
     1. Select the HTTP Artifact checkbox under Single Sign-On Service.
     2. Make any optional configurations.
        For more information, see the following:
        ○ Disabling IdP-Initiated and SP-Initiated SSO and SLO
        ○ Configuring the Lifetime of Identity Provider Sessions
        ○ Configuring Identity Providers as Proxies
   ○ To accept artifacts for Single Log-Out (SLO):
     1. Select the HTTP Artifact checkbox under Single Log-Out Service.
     2. Make any optional configurations.
        For more information, see the following:
        ○ Disabling IdP-Initiated and SP-Initiated SSO and SLO
        ○ Determining the Channel Used for SLO by the Identity Provider

3. Configuring the Endpoints for the Trusted Service Provider
   With this procedure you configure the outgoing connection to the identity provider. This procedure assumes that you have already trusted a service provider.
For more information about trusting a service provider, see Adding Service Providers.

a. Choose Trusted Providers.
b. Select a service provider and choose the Edit pushbutton.
c. Choose the Endpoints tab.
d. Configure the Assertion Consumer Endpoints, Single Log-Out Endpoints, and Artifact Endpoints to use HTTP Artifact and SOAP bindings as required.
   1. Add HTTP artifact bindings.
   2. Enter an index value for the endpoint.
   3. Enter the endpoint URLs for the services on the service provider.
e. Save your entries.

4. Configuring the Service Provider
a. Check that the service provider endpoints are configured to accept HTTP artifact bindings from the identity provider.
b. Check that the service provider is configured to use HTTP artifact bindings to connect to the endpoints of the identity provider.
c. Consider disabling front-channel communication bindings for the service provider endpoints. If the identity provider only accepts back-channel communications, there is no reason to expose the endpoint to front-channel bindings.

For more information about how to configure the service provider, see the documentation of your service provider.

Related Information

Disabling IdP-Initiated and SP-Initiated SSO and SLO [page 73]
Configuring the Lifetime of Identity Provider Sessions [page 71]
Configuring Identity Providers as Proxies [page 55]
Disabling IdP-Initiated and SP-Initiated SSO and SLO [page 73]
Determining the Channel Used for SLO by the Identity Provider [page 76]

1.3.4.3 Enabling Back-Channel Communication with SOAP

Use this procedure to configure the back-channel parameters for SOAP.

Procedure

1. Accept SOAP Bindings
   a. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
   b. Choose SAML 2.0 Local Provider.
   c. Choose the Identity Provider Settings tab.
1. Under **Single Log-Out Service**, select the **SOAP** checkbox.

**Note**
When the SOAP binding is enabled, the user session persists in the database.

2. Configure the Endpoints for the Trusted Identity Provider

   With this procedure you configure the outgoing connection to the service provider. This procedure assumes that you have already trusted a service provider.

   For more information about trusting a service provider, see **Trusting Service Providers**.

   a. Choose **Trusted Providers**.
   b. Select a service provider and choose the **Edit** pushbutton.
   c. Choose the **Endpoints** tab.
   d. Configure the **Single Log-Out Endpoints** to use SOAP binding.

      1. Add the SOAP binding.
      2. Configure and select a destination to use from the destination service of the AS Java. The destination must include the endpoint URL for the SLO service on the service provider. If the service provider requires authentication for SOAP, you must also configure the destination to use the required user.

         **Note**
         After configuring the endpoints, you can change the URL by editing the Location URL field. This updates the URL used in the destination service.

   3. Determine if you want the log-out response sent to a different URL.

      If yes, enter the custom response location in the **Response Location URL** column.

   e. Save your entries.

3. Configure the Service Provider

   a. Check that the service provider endpoints are configured to accept HTTP artifact and SOAP bindings from the identity provider.
   b. Check that the service provider is configured to use HTTP artifact and SOAP bindings to connect to the endpoints of the identity provider.
   c. Consider disabling front-channel communication bindings for the service provider endpoints. If the service provider only accepts back-channel communications, there is no reason to expose the endpoints to front-channel bindings.

   For more information about how to configure the service provider, see the documentation of your service provider.

### 1.3.5 Configuring Front-Channel Communication

Front-channel communication uses HTTP POST or HTTP redirect bindings over the client between the service provider and the identity provider. Use front-channel bindings when response time to the client request is more important than ensuring that SAML messages are not exposed to the client or any malicious third-parties. Back-channel communication increases the number of messages the service provider and identity provider must exchange to log on.
Prerequisites

- You have determined which front-channel bindings you want to support.

<table>
<thead>
<tr>
<th>Binding</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP POST</td>
<td>Transports SAML messages in the body of the message. There are no length limitations. See disadvantages of HTTP redirect below.</td>
<td>○ There may be some clients that do not support HTTP POST. ○ To avoid user interaction to send the client from one server to the next, clients employ an auto post function. The auto post function uses JavaScript. Depending on your situation, the use of JavaScript can represent a security risk.</td>
</tr>
<tr>
<td>HTTP Redirect</td>
<td>Client sent from one server to the next without interaction from the user.</td>
<td>Redirect transports the SAML message in the URL. If the URL is too long, the client truncates the URL. If you use long URLs or include security options such as encryption of message elements, avoid HTTP redirect.</td>
</tr>
</tbody>
</table>

- SAML 2.0 has been enabled on your SAP NetWeaver Application Server (AS) Java. For more information, see Enabling the SAML Identity Provider.

Related Information

Enabling the SAML Identity Provider [page 29]

1.3.5.1 Disabling Front-Channel Communication

Use this procedure to restrict authentication to back-channel communication.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose [SAML 2.0][Local Provider].
3. Choose the Identity Provider Settings tab.
4. For the Single Sign-On and Single Log-Out (SLO) services, deselect the HTTP POST and HTTP Redirect checkboxes.
5. Disable HTTP POST and HTTP redirect bindings from trusted service providers.
For more information, see the product documentation for your service provider.

1.3.5.2 Enabling Front-Channel Communication

Use this procedure to accept front-channel communication and configure the other front-channel parameters.

Procedure

1. Determine which services accept front-channel communication
   a. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
   b. Choose SAML 2.0 Local Provider
   c. Choose the Identity Provider Settings tab.
   d. Determine for which services you want to accept front-channel communication from service providers.
      For Single Sign-On (SSO):
      1. Select the HTTP POST or HTTP Redirect checkboxes under Single Sign-On Service.
      2. Make any optional configurations.
         For more information, see the following:
         ○ Disabling IdP-Initiated and SP-Initiated SSO and SLO
         ○ Configuring the Lifetime of Identity Provider Sessions
         ○ Configuring Identity Providers as Proxies
      For Single Log-Out (SLO):
      1. Select the HTTP POST or HTTP Redirect checkboxes under Single Log-Out Service.
      2. Make any optional configurations.
         For more information, see the following:
         ○ Disabling IdP-Initiated and SP-Initiated SSO and SLO
         ○ Determining the Channel Used for SLO by the Identity Provider

2. Configure the endpoints for the trusted service provider
   With this procedure you configure the outgoing connection to the service provider. This procedure assumes that you have already trusted a service provider.
   For more information about trusting a service provider, see Adding Service Providers.
   a. Choose Trusted Providers.
   b. Select a service provider and choose the Edit pushbutton.
   c. Choose the Endpoints tab.
   d. Configure any Assertion Consumer Endpoints and Single Log-Out Endpoints.
      1. Add any HTTP POST and HTTP redirect bindings.
      2. Set an index value as required.
      3. Enter the endpoint URLs for the services on the service provider.
   e. Save your entries.

3. Configure the service provider.
   a. Check that the service provider endpoints are configured to accept HTTP POST or HTTP redirect from the identity provider.
b. Check that the service provider is configured to use HTTP POST or HTTP redirect to connect to the endpoints of the identity provider.

c. Consider disabling back-channel communication bindings for the service provider endpoints.
   If the service provider only accepts front-channel communications, there is no reason to expose the endpoint to back-channel bindings.
   For more information about how to configure the service provider, see the documentation of your service provider.

Related Information

Disabling IdP-Initiated and SP-Initiated SSO and SLO [page 73]
Configuring the Lifetime of Identity Provider Sessions [page 71]
Configuring Identity Providers as Proxies [page 55]
Disabling IdP-Initiated and SP-Initiated SSO and SLO [page 73]
Determining the Channel Used for SLO by the Identity Provider [page 76]
Adding Service Providers [page 39]

1.3.6 Configuring Support for Enhanced Client or Proxy

Prerequisites

The ECP knows or is capable of discovering which identity provider the service provider trusts.

Context

The Enhanced Client or Proxy (ECP) profile of the SAML 2.0 specification is useful in the following situations:

- You have a client with extended capabilities and you want the client to take on more responsibility in the exchange. For example, the client can determine the appropriate identity provider.
- Your client has limited capabilities so you delegate some of these tasks to an enhanced proxy. For example, a wireless access point (WAP).
- You cannot use other bindings. Some possible examples are as follows:
  - The client does not support redirects.
  - The client does not support JavaScript, preventing auto form post.
  - A firewall prevents the identity provider and service provider from communicating directly, preventing the artifact binding.

The ECP profile enables the client to contact the identity provider with the authentication request generated by the service provider. Exchanges between the ECP and the identity provider use SOAP.
Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 Local Provider.
3. Choose the Identity Provider Settings tab.
5. Save your entries.
6. Configure the service provider to support the reverse SOAP (PAOS) binding.
   For more information, see the documentation supplied by the service provider vendor.

1.3.7 Trusting Service Providers

Trust ing service providers is a two-step process. First, add a service provider to the list of trusted service providers. This involves adding connection information as well as the exchange of public keys for encryption and digital signatures and identification of the endpoints. Second, configure federation for the provider. The federation configuration identifies the identity information that the service provider requires.

1.3.7.1 Adding Service Providers

The identity provider provides identity information to service providers for applications that the service providers protect. An identity provider can only do this for service providers in its list of trusted service providers. Use this procedure to add a service provider to the list of trusted service providers.

Prerequisites

- You have configured a service provider in your network.
- If you intend to add the service provider manually (without using a metadata XML file), you have imported the public-key certificates of the service provider for encryption and digital signature of SAML messages. Import these certificates into the key storage of the SAP NetWeaver Application Server (AS) Java.
- If you intend to add the service provider from a metadata file, you have a means of accessing the metadata of the provider from a secure source.
  - If you upload the metadata from a file, we assume that you received the file from a trustworthy source. The identity provider accepts the metadata. However, if the metadata is signed by the service provider, the identity provider checks that the issuer of the certificate of the signer is trusted by the AS Java. If the AS Java does not trust the issuer, the identity provider rejects the metadata.
  - If you upload the metadata from a URL, the identity provider distinguishes between accessing the URL with HTTP or HTTPS in addition to whether the metadata is signed or not.
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Metadata is Signed</th>
<th>Metadata is Unsigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>If the issuer of the signing certificate is trusted, the identity provider accepts the metadata.</td>
<td>The identity provider rejects the metadata. There is no way for the identity provider to verify the source of the metadata.</td>
</tr>
<tr>
<td>HTTPS</td>
<td>If the issuer of the signing certificate is trusted, the identity provider accepts the metadata. As an additional check, you can require the identity provider to check if the issuer of the server certificate for Secure Sockets Layer (SSL) is trusted. If the issuer is not trusted, the service provider rejects the metadata.</td>
<td>If the issuer of the server certificate for SSL is trusted, the identity provider accepts the metadata.</td>
</tr>
</tbody>
</table>

### Procedure

1. Start SAP NetWeaver Administrator with the quick link `/nwa/auth`.
2. Choose **SAML 2.0** > **Trusted Providers**.
3. From the list of trusted providers, show the service providers.
4. Choose the **Add** pushbutton and choose one of the following:
   - **Manually**
   - **Specifying Metadata URL**
     
     Provide the URL of the metadata XML file of the service provider and determine if you want to verify the SSL server certificate of the service provider.
     - If the metadata is unsigned and you are accessing the URL with HTTPS, select the **Verify SSL Peer Identity** checkbox. Otherwise the identity provider rejects the metadata. To view the certificates of the certificate authorities the AS Java trusts, choose the **Trusted Issuers** pushbutton.
     
     For more information about configuring the trusted issuers, see **Selecting the Keystore View for SSL for the Identity Provider**.
     - If the metadata is signed and you are accessing the URL with HTTPS, you can select the **Verify SSL Peer Identity** checkbox as an option to confirm the identity of the service provider.
     - If you are accessing the URL with HTTP, clear the **Verify SSL Peer Identity** checkbox.
   - **Uploading Metadata File**
     
     Provide the path to the metadata XML file of the service provider.
5. Enter a name and an alias.

### Caution

Do not change the name of the service provider if the metadata XML file provides it. The name must match the name configured in the service provider **exactly**.

6. Enter the required data for digital signatures and encryption.
**Note**
The identity provider automatically sets SHA-1 digest algorithm for signing the outgoing SAML 2.0 messages. If you want to change the digest algorithm to SHA-256, you have to do it later under the Signature and Encryption tab.

**Caution**
For this configuration to work, the service provider has to support the same signing digest algorithm.

a. Select the public-key certificates for checking the digital signature of the service provider and encrypting messages sent to the service provider from the key storage.

   If you add the service provider from a metadata XML file, the public-key certificates are already configured.

b. Choose an encryption algorithm.

   **Note**
The cryptographic suite of the service provider must support the encryption algorithm you choose or it cannot decrypt your messages.

c. Choose the signature and encryption options for requests, responses, and assertions for Single Sign-On (SSO), Single Log-Out (SLO), and artifact resolution.

   The signature and encryption options must match with those of the service provider. If the service provider requires that SAML assertions are always digitally signed and the identity provider never signs them, then the SAML configuration cannot function.

→ **Recommendation**

   Give some thought to your encryption and signature options and make choices that make sense for your configuration. These also depend on the environment your SAML network is working in. Systems that operate in a secured area behind a firewall have different requirements from systems exposed to the Internet. We have the following recommendations:

   ○ Encryption

      Encrypt or require encryption for those elements that can expose authentication or other personal data about the users. If you use the transient or persistent name ID formats, these name IDs are already opaque. There is no need to encrypt these name IDs. The e-mail name ID format, however, can reveal the user’s real name and contact information.

      When using the transient and persistent name ID formats, you can send attributes. These attributes can also reveal personal information, which you should encrypt.

   ○ Digital Signatures

      The SAML standard provides many points in the process at which you can sign and check for signatures. Do this only where it makes sense. For example, you can sign the SAML assertion and the SAML response. It does not make sense for the identity provider to sign the SAML response and then pack it in a SAML assertion and sign it again before sending the assertion to the service provider. This would only make sense if you developed a custom process to separate the SAML response from the SAML assertion and send the response through a third party before the response is processed. You can further complicate the process by using the HTTP artifact binding and signing the artifact response. In this case, the identity provider would sign the message three times.

      SAP’s service provider supports signature inheritance. If the SAML 2 response is signed, the service provider considers the SAML 2 assertion to be signed. Likewise, if the SAML 2 artifact
response is signed, the service providers considers the SAML 2 response and SAML 2 assertions it contains to be signed.

d. Enter the required data for the SSO, SLO, and artifact resolution service (ARS) endpoints. The metadata XML provides the bindings supported by the service provider. If you add new bindings, you must configure the service provider to support them.
e. Choose the Finish pushbutton.

Related Information

Selecting the Keystore View for SSL for the Identity Provider [page 44]

1.3.7.2 Updating the Configuration of Trusted Providers

When you make changes to the configuration of a trusted provider, you must update the configuration of the trust relationship to match your changes.

Prerequisites

You have a means of accessing the metadata of the provider from a secure source.

- If you upload the metadata from a file, we assume that you got the file from a trustworthy source. The identity provider accepts the metadata. However, if the metadata is signed by the service provider, the identity provider checks that the issuer of the certificate of the signer is trusted by the SAP NetWeaver Application Server (AS) Java. If the AS Java does not trust the issuer, the identity provider rejects the metadata.
- If you upload the metadata from a URL, the identity provider distinguishes between accessing the URL with HTTP or HTTPS in addition to whether the metadata is signed or not.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Metadata is Signed</th>
<th>Metadata is Unsigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>If the issuer of the signing certificate is trusted, the identity provider accepts the metadata.</td>
<td>The identity provider rejects the metadata. There is no way for the identity provider to verify the source of the metadata.</td>
</tr>
</tbody>
</table>
Protocol | Metadata is Signed | Metadata is Unsigned
--- | --- | ---
HTTPS | If the issuer of the signing certificate is trusted, the identity provider accepts the metadata. As an additional check, you can require the identity provider to check if the issuer of the server certificate for Secure Sockets Layer (SSL) is trusted. If the issuer is not trusted, the identity provider rejects the metadata. | If the issuer of the server certificate for SSL is trusted, the identity provider accepts the metadata.

**Context**

The following is a list of changes that require an update of the trusted provider configuration:

- **New certificates for digital signature or encryption**
  You can have a primary and secondary certificate for signatures and encryption. This enables you to span the time when an old certificate is due to expire and you have not yet configured all peers to accept the new one.

- **Changed signature or encryption options**

- **Changed Assertion Consumer Service, Single Log-Out, or Artifact Resolution Service endpoints**

**Procedure**

1. Start SAP NetWeaver Administrator with `/nwa/auth`.
2. Choose [SAML 2.0 > Trusted Providers](#)
3. From the list of trusted providers, show the service providers.
4. Select a service provider.
5. Choose the [Update] pushbutton and choose one of the following:
   - Specifying Metadata URL
     Provide the URL of the metadata XML file of the service provider and determine if you want to verify the SSL server certificate of the service provider.
   - If the metadata is unsigned and you are accessing the URL with HTTPS, select the [Verify SSL Peer Identity](#) checkbox. Otherwise the identity provider rejects the metadata. To view the certificates of the certificate authorities that the AS Java trusts, choose the [Trusted Issuers](#) pushbutton. For more information about configuring the trusted issuers, see [Selecting the Keystore View for SSL for the Identity Provider](#).
   - If the metadata is signed and you are accessing the URL with HTTPS, you can select the [Verify SSL Peer Identity](#) checkbox as an option to confirm the identity of the service provider.
   - If you are accessing the URL with HTTP, clear the [Verify SSL Peer Identity](#) checkbox.
   - Uploading Metadata File
     Provide the path to the metadata XML file of the service provider.
6. Follow the instructions in the wizard to update the configuration.
Related Information

Selecting the Keystore View for SSL for the Identity Provider [page 44]

1.3.7.3 Selecting the Keystore View for SSL for the Identity Provider

The SAML identity provider stores the client certificates for trusted service providers for the establishment of Secure Sockets Layer (SSL) connections in a keystore view of the keystore service.

Context

The identity provider uses these certificates when getting the metadata of trusted service providers over HTTPS. Use this procedure to determine the keystore view.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 Local Provider.
3. Choose the Edit pushbutton.
4. Choose the General Settings tab.
5. Under Signature and Encryption, enter a keystore view in the Trusted CAs Keystore View field.
6. Save your entries.

Related Information

Adding Service Providers [page 39]
Updating the Configuration of Trusted Providers [page 42]

1.3.7.4 Configuring Out-of-Band Account Linking

The service provider defines which name ID format it requires in the SAML authentication request that it forwards to the identity provider. As long as the identity provider supports this name ID format, it returns the requested information in the SAML response, including any attributes. Identity federation is the mapping of the
requested information to the information provided by the identity provider. Without this mapping, no federation can exist.

**Prerequisites**

You have trusted a service provider. For more information, see *Adding Service Providers*.

**Procedure**

1. Start SAP NetWeaver Administrator with the quick link `/nwa/auth`.
2. Choose `SAML 2.0 > Trusted Providers`.
3. Select a service provider and choose the `Edit` pushbutton.
4. On the *Identity Federation* tab, choose the `Add` pushbutton.
5. Choose a name ID format and source.

   If the source is a user attribute, some name ID formats enable you to configure it. To make this attribute viewable in identity management, add it as a custom attribute.

   The service provider requests the name ID format from the trusted identity provider. After the identity provider authenticates the user, the identity provider uses the source to determine where it gets the name ID to put in the SAML response. The service provider then uses the name ID to identify the user.

### Name ID Formats for Out-of-Band Account Linking

<table>
<thead>
<tr>
<th>Name ID Format</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail</td>
<td>User attribute</td>
<td>Provides the e-mail address of the authenticated user</td>
</tr>
<tr>
<td>Kerberos</td>
<td>ADS data source</td>
<td>Provides the Kerberos Principal Name (KPN) and realm of the authenticated user from the ADS data source of the user management engine (UME)</td>
</tr>
<tr>
<td>JAAS Subject</td>
<td></td>
<td>Provides the KPN and realm of the authenticated user from the Java Authentication and Authorization Service (JAAS) subject</td>
</tr>
<tr>
<td>User attribute</td>
<td></td>
<td>Provides the KPN and realm of the authenticated user from a configurable user attribute</td>
</tr>
<tr>
<td>Name ID Format</td>
<td>Source</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Persistent</td>
<td>User attribute</td>
<td>Provides the persistent name ID of the authenticated user from a configurable user attribute.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>&lt;br&gt;The <em>Persistent</em> name ID format allows for other configuration options when not using out-of-band account linking. For more information, see <a href="#">Configuring Identity Federation with Persistent Pseudonyms</a>.</td>
</tr>
<tr>
<td>Unspecified</td>
<td>Logon ID</td>
<td>Provides the logon ID of the authenticated user</td>
</tr>
<tr>
<td></td>
<td>Logon alias</td>
<td>Provides the logon alias of the authenticated user</td>
</tr>
<tr>
<td></td>
<td>User attribute</td>
<td>Provides the value from a configurable user attribute of the authenticated user</td>
</tr>
<tr>
<td>Windows Name</td>
<td>ADS data source</td>
<td>Provides the Microsoft Windows qualified domain name of the authenticated user from the ADS data source of the UME</td>
</tr>
<tr>
<td></td>
<td>User attribute</td>
<td>Provides the domain-qualified Microsoft Windows name of the authenticated user from a configurable user attribute</td>
</tr>
<tr>
<td>X509 Subject Name</td>
<td>User attribute</td>
<td>Provides the subject name of the authenticated user from a configurable user attribute</td>
</tr>
</tbody>
</table>

6. Create a mapping between the SAML 2.0 attributes and the user management engine (UME) attributes to send with the SAML assertion to the service provider.

Negotiate with the administrator of the service provider which data goes into the SAML 2.0 attributes, and how you name the attributes. The service provider interprets the SAML 2.0 attributes to determine what attributes the user has on the service provider. The SAML 2.0 attributes can be configured with constant values, and you can set these attributes on the *Default User Attributes* tab. Make sure you specify the name and the value of each attribute. The SAML 2.0 attributes can also refer to a user profile, and you can set these attributes on the *User-Based Assertion Attributes* tab, or to a role membership or a group membership, for which you can set attributes on the *Authorization-Based Assertion Attributes* tab. The way in which the service provider interprets these attributes is dependent on the configuration and implementation of the SAML 2.0 service provider. For example, if you configure the authorization-based attribute `memberOf` with the groups `Administrators`, `Translators`, and `Developers`, and the user is a member of the groups `Administrators`, `Translators`, and `Testers`, then the identity provider sends a SAML assertion to the service provider with an attribute `memberOf` and values `Administrators` and `Translators`.

For more information, see the documentation supplied by the service provider vendor.
7. Save your entries.
8. Configure the service provider to use the same name ID format.
   For more information about configuring a service provider, see the documentation supplied by the service provider vendor.

▶ Note

Donna Moore has configured her service provider to require the E-mail name ID format. A trusted identity provider sends her service provider a SAML response with Lauren.Becker@example.com as the subject. The service provider searches for a user with that value as an e-mail address. If the result is a single user, log-on succeeds.

Laurent Becker has a different user ID on the service provider and the identity provider, but his e-mail address is the same in both systems. A simple mapping would be to have the identity provider use the E-mail name ID format, too.

Imagine that the identity provider uses the e-mail address for the user ID and does not use an attribute for e-mail. Then the identity provider would use the Unspecified name ID format to return the user ID. Donna must reconfigure her service provider to match. If the identity provider cannot support the E-mail name ID format, Donna must configure the service provider to request the Unspecified name ID format and select the e-mail user attribute as the source.

Related Information

Adding Service Providers [page 39]
Configuring Identity Federation with Persistent Pseudonyms [page 49]
Adding Custom User Attributes for SAML [page 51]

1.3.7.5 Configuring Identity Federation with Transient Users

Identity federation using transient pseudonymous identifiers enables service providers to provide authenticated users with access to their systems without needing to know specific details about those users. You negotiate with the administrators of service providers to determine what kind of SAML 2.0 attributes they require. They determine how these attributes are mapped to user attributes, groups, and roles in their systems, while you handle the management of the users and their authentication. By managing SAML 2.0 attributes on your system, you can determine what access the users have on service providers’ systems, without intervention by the service providers’ administrators.
Prerequisites

You have trusted a service provider. For more information, see Adding Service Providers.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 > Trusted Providers.
3. Select a service provider and choose the Edit pushbutton.
4. On the Identity Federation tab, choose the Add pushbutton.
5. Enter the name ID format Transient.
6. Create a mapping between the SAML 2.0 attributes and the user management engine (UME) attributes to send with the SAML assertion to the service provider.
   
   You negotiate with the administrator of the service provider which data goes into the SAML 2.0 attributes and how you name the attributes. The service provider interprets the SAML 2.0 attributes to determine what attributes the transient user has on the service provider. The SAML 2.0 attributes can be configured with constant values, and you can set these attributes on the Default User Attributes tab. Make sure you specify the name and the value of each attribute. The SAML 2.0 attributes can also refer to a user profile, and you can set these attributes on the User-Based Assertion Attributes tab, or to a role membership or a group membership, for which you can set attributes on the Authorization-Based Assertion Attributes tab.
   
   The way in which the service provider interprets these attributes is dependent on the configuration and implementation of the SAML 2.0 service provider. For example, if you configure the authorization-based attributememberOf with the groups Administrators, Translators, and Developers, and the user is a member of the groups Administrators, Translators, and Testers, then the identity provider sends a SAML assertion to the service provider with an attribute memberOf and values Administrators and Translators.

   For more information, see the documentation supplied by the service provider vendor.

7. Save your entries.
8. Configure the service provider to accept the transient name ID format and map the SAML 2.0 attributes to transient users.

   For more information about configuring a service provider, see the documentation supplied by the service provider vendor.

Related Information

Adding Service Providers [page 39]
Adding Custom User Attributes for SAML [page 51]
1.3.7.6 Configuring Identity Federation with Persistent Pseudonyms

Use this procedure to enable identity federation when no previous linking between the accounts exists. Once authenticated by the identity provider, the service provider can enable users to link their accounts interactively or the service provider can automatically create a federated account with SAML 2.0 attributes supplied by the identity provider. If the accounts are already linked, log-on occurs with the persistent name ID.

Prerequisites

- You have trusted a service provider.
  For more information, see Adding Service Providers.
- You have determined whether the service provider requires SAML attributes for automatic account creation.
  If the service provider is configured to support automatic account creation, the service provider uses SAML 2.0 attributes and values sent by the identity provider to create user accounts. To support this option, you must negotiate with the administrator of the service provider to determine what data the service provider requires and how to name the SAML 2.0 attributes carrying the data. Configure the identity provider to supply these attributes when configuring identity federation with persistent pseudonyms.

  **Note**
  To use custom user management engine (UME) attributes with SAML 2.0 attributes, you need to add them for SAML on the identity provider. For more information, see Adding Custom User Attributes for SAML.

Context

You can also use out-of-band account linking with persistent pseudonyms, but the linking must be established ahead of time. For more information, see Configuring Out-of-Band Account Linking.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 >> Trusted Providers >>
3. Select a service provider and choose the Edit pushbutton.
4. On the Identity Federation tab, choose the Add pushbutton.
5. Enter the name ID format Persistent.
6. Configure the user attribute.
7. Create mappings for automatic account creation as required.
   a. Allow the identity provider to issue assertion attributes with constant values. You can set the mapping of attributes with constant values by choosing the Default User Attributes tab. To add a new SAML2 Attribute, choose the Edit button and then the Add button. Specify the name and the value.
   b. Create a mapping between the SAML 2.0 attributes and the UME attributes of the service provider based on the information entered by the user. You can set these attributes by choosing the User-Based Assertion Attributes tab.
   c. Map any group or role memberships to SAML 2.0 attributes. You can set this configuration by choosing the Authorization-Based Assertion Attributes tab.

Example
The table below shows a mapping between SAML 2.0 attributes and the UME attributes. Included are the values for a user named Laurent Becker. When the identity provider sends a SAML 2.0 authentication response, it includes the values of the mapped UME attributes as SAML 2.0 attributes.

<table>
<thead>
<tr>
<th>SAML 2.0 Attribute</th>
<th>Value</th>
<th>UME Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st-name</td>
<td>Laurent</td>
<td>firstname</td>
</tr>
<tr>
<td>2nd-name</td>
<td>Becker</td>
<td>lastname</td>
</tr>
<tr>
<td>mail</td>
<td><a href="mailto:Laurent.Becker@example.com">Laurent.Becker@example.com</a></td>
<td>email</td>
</tr>
</tbody>
</table>

8. Save your entries.
9. Configure the service provider to accept the persistent name ID format and any attributes required by your configuration.
   For more information about configuring a service provider, see the documentation supplied by the service provider vendor.

Results

- At the request of the service provider, the identity provider can create a persistent name ID, if none exists for the user account on the identity provider. If the service provider forbids the identity provider to create a persistent name ID, you must ensure that all users already have a persistent name ID configured on the identity provider before they log on to the service provider.
- For automatic account creation, the service provider can specify that specific SAML 2.0 attributes are required. For automatic account creation to succeed, you must ensure that those attributes are populated for all users.

Note
Donna Moore has recently configured her network to support SAML 2.0. The users are still logging in to each system with a separate user ID and password. Donna has set up a new identity provider with all users and assigned each one a persistent name ID. She has just upgraded her legacy systems to support SAML 2.0 as service providers. In each system, she trusts the SAML 2.0 identity provider and requires the Persistent name ID format. Since all the users already know their passwords in each system, she enables
interactive account linking. Whenever a user logs on to a system for the first time since conversion, the user enters his or her log-on information and the service provider adds the persistent name ID from the identity provider to the local account. Donna does not need to go through the laborious process of adding the persistent ID to every account in every system. The users do it themselves.

Related Information

Configuring Out-of-Band Account Linking [page 44]
Adding Service Providers [page 39]
Adding Custom User Attributes for SAML [page 51]

1.3.7.7 Trust When the Host is Service Provider and Identity Provider

The following applies to the trust relationship between a service provider and an identity provider enabled on the same SAP NetWeaver Application Server (AS) Java:

- Trust is implicit between the identity provider and the service provider. You do not need to configure trust between the two SAML 2.0 entities.

- You cannot configure endpoints and name ID formats in the trust relationship. The providers share endpoints. Every user authenticated on the identity provider is automatically authenticated for the service provider, because the entities have the same user store and authentication framework.

- The service provider can trust identity providers in addition to the identity provider on the host.

1.3.7.8 Adding Custom User Attributes for SAML

To map user attributes that are not part of the user profile by default to SAML attributes in a SAML 2.0 authentication response, you need to add them to the system. The identity provider uses these attributes in all kinds of federations.

Context

For more information, see the following:

- Configuring Out-of-Band Account Linking
- Configuring Identity Federation with Persistent Pseudonyms
- Configuring Identity Federation with Transient Users
Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 Local Provider.
3. Choose the Edit pushbutton.
4. Choose the User Attributes tab.
5. Choose the Add pushbutton.
6. Enter data as required.
7. Save your entries.

Related Information

Other applications can define custom attributes, too. You can also make existing custom attributes available for the SAML configuration. To identify the technical names of custom attributes from other applications, try the following:

- If the identity management user interface has been configured to support management of the custom attribute, refer to the user management engine (UME) configuration.
- The attribute may be defined in the UME data source configuration file.
- Contact the application developer or refer to the application documentation or source code.

Configuring Out-of-Band Account Linking [page 44]
Configuring Identity Federation with Transient Users [page 47]
Configuring Identity Federation with Persistent Pseudonyms [page 49]

1.4 Optional Configurations

This section describes configuration options you can use to optimize or improve the operation of the identity provider.

1.4.1 Adding Custom Authentication Contexts

If the service provider offers an authentication context for authentication requests that is not part of the standard configuration of the identity provider, you can add the authentication context as a custom
authentication context. Your service provider can then request that the identity provider use this context to authenticate users.

**Prerequisites**

You know the name of the authentication context from the operator of the service provider.

**Procedure**

1. Start the SAML 2.0 configuration application (transaction `SAML2`).
2. On the **Local Provider** tab, choose the **Authentication Contexts** tab.
3. Choose the **Edit** pushbutton.
4. Choose the **Add** pushbutton.
5. Enter a display name for the authentication context in the **Alias** field.
6. Enter the name of the authentication context as it is required by the service provider in the **Name** field.
   When required, the service provider enters the name of the authentication context in the authentication request.
7. Save your entries.
8. Map the new authentication context to a log-in module.
   For more information, see **Mapping Authentication Contexts to Log-in Modules**.

**Related Information**

Mapping Authentication Contexts to Log-in Modules [page 77]

**1.4.2 Configuring the Identity Provider for Discovery With CDCs**

Different applications on a service provider can require different identity providers. A service provider requires the means to discover which identity provider it should use. One way is to use a common domain cookie (CDC). A CDC stores a list of the identity providers recently visited. A write service for the common domain cookie is available on the identity provider side. It allows the identity provider to append itself to the list of identity providers stored in the CDC. At the opposite side, the service provider has a read service for the CDC. It allows the service provider to read and use the CDC identity providers list.
The identity provider can use an internal write service, an external write service, or both. An internal CDC write service is one that is located in the same domain as the identity provider. An external CDC write service is one located in a different domain. If you are using an external CDC service, you need to configure:

- The list of sites the CDC service is allowed to redirect to. You also have to add the identity provider to this list.
- The list of identity providers stored in the CDC. You also have to add the identity provider to this list.

For the internal CDC write service, this is not necessary. For more information, see Common Domain and Identity Provider Discovery.

### Write Service of AS Java

Every SAP NetWeaver Application Server (AS) Java 7.2 and later includes a read and write CDC service. As an external service you can use the following URL: https://<hostname>:<port>/saml2/idpdiscovery/write

### Related Information

Common Domain and Identity Provider Discovery [page 14]

### 1.4.2.1 Defining the CDC Write Service to be Used by the Identity Provider

#### Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 Local Provider.
3. Choose the Edit pushbutton.
4. Choose the Identity Provider Settings tab.
5. Specify whether you want to enable an internal or external CDC write service. You can enable both at the same time.
   - If you enable access to the CDC internal service, determine when the cookie expires and relax the domain until the identity provider and service provider share a common domain.
   - If you enable access to the CDC external service, enter the URL of the service, including the fully qualified domain name. Configure the external service as required.

If you enable an external CDC write service, you need to specify a valid CDC write service URL.
1.4.2.2 Configuring the External CDC Write Service

Procedure

1. Open the CDC write service configuration window. This can be done in one of the following ways:
   ○ Directly in your Web browser
     Use the following URL: https://<hostname>:<port>/webdynpro/resources/sap.com/tc-sec-saml2-cfg-wd/IdPDiscConfig?idp=<idp_name>
   ○ Using the SAML 2.0 configuration interface
     Open the SAML 2.0 configuration user interface, and choose Local Provider ➤ Identity Provider Settings ➤ Identity Provider Discovery: Common Domain Cookie (CDC) ➤ External CDC Write Service URL ➤ Configure External CDC Write Service.

2. Choose Edit.

3. Add the identity provider’s site to the Allowed Redirect Sites tab for the CDC write service.
   Use the format <protocol>://<hostname>:<port>.

4. Add the identity provider to the list of identity providers allowed in the common domain cookie.

5. Save your entries.

1.4.3 Configuring Identity Providers as Proxies

An identity provider functioning as a proxy can ease the management of configuring trust when there is a large number of identity providers and service providers in your SAML 2.0 landscape.

Prerequisites

- Your identity provider is also a service provider.
  For more information, see Enabling the SAML Identity Provider.
- You have one or more identity providers to serve as authenticating identity providers in your SAML 2.0 landscape.
- You have one or more service providers in your SAML 2.0 landscape.

Procedure

1. Configure trust.
   ○ The proxying identity provider trusts service providers in your SAML 2.0 landscape.
     For more information, see Adding Service Providers.
   ○ The service provider of the proxying identity provider trusts authenticating identity providers in your SAML 2.0 landscape.
For more information, see the documentation for trusting identity providers on the SAP Help Portal for SAP NetWeaver Application Server (AS) Java.

2. Start SAP NetWeaver Administrator with the quick link /nwa/auth.

3. Choose SAML 2.0 Local Provider.

4. Choose the Proxying Settings tab.

5. Choose the Edit pushbutton.

6. Determine if you want to enforce proxying.

   Unless another identity provider is specified in the authentication request or, in the case of IdP-initiated Single Sign-On (SSO), in the URL parameter saml2idp, the proxying identity provider attempts to authenticate the user. To prevent the identity provider from authenticating the user, you can enforce proxying.

   To enforce proxying, enter Yes in the Enforce Proxying field under the Proxying Settings tab.

   Do not use this option if you want this identity provider to authenticate users for service providers.

7. Enter a proxy count restriction for the identity provider in the Proxy Count (Assertion) field.

   The identity provider uses the proxy count restriction if it does not receive a proxy count value in the authentication response. The identity provider then inserts this value in the assertion it hands down the chain.

8. Select Operational Mode: Service Provider or Operational Mode: Identity Provider and Service Provider.

9. Enter a proxy count restriction for the service provider in the Proxy Count (AuthnRequest) field.

   The identity provider uses the proxy count restriction if it does not receive a proxy count value in the authentication request. The identity provider then inserts this value in the request it hands up the chain.

10. Set the configuration for the scoping element.

    For more information, see Principles and Configuration of the Scoping Element.

    i Note

    The system sets the option to send the scoping element by default.

11. Save your entries.

Related Information

   Enabling the SAML Identity Provider [page 29]
   Adding Service Providers [page 39]
   Principles and Configuration of the Scoping Element [page 79]

1.4.4 Configuring Identity Provider Extensions

You can extend the current functionality of the identity provider (IdP) with custom extension plug-ins such as external adapters and attribute providers. The external adapters can influence the IdP’s authentication process by selecting custom authentication contexts at runtime and by making decisions on whether to create a SAML
2.0 assertion. After a successful authentication, the attribute providers supply custom attributes to the generated SAML 2.0 assertion.

To configure the extensions, you have to do the following:

1. To specify a custom extension, you have to create a Java EE application and implement an extension interface.
2. To make the custom extension available for the IdP, deploy the application.
3. For the IdP to use an extension, you have to configure it in SAP NetWeaver Administrator. There are some system requirements what AS Java versions the IdP extensions are available for. For more details, see Related Information.

Related Information

System Requirements [page 23]
Implement an Identity Provider Extension [page 57]
Configure SAP NetWeaver Administrator [page 64]

1.4.4.1 Implement an Identity Provider Extension

Prerequisites

To implement an extension, you need to install an Eclipse-based tool that allows you to develop J2EE Web applications. We advise you to use SAP NetWeaver Developer Studio. This section provides guidelines about configurations in the Developer Studio. For more information about where to install it from, see Developer Studio Download Links.

Procedure

1. Create a Web module project.
   a. Create an enterprise application project as you add a Web module to the project.
      This operation creates another Web module project.
   b. Add a runtime application reference to the saml2_lib application.
      To do this, add the following reference in the application-j2ee-engine.xml file:

      ```xml
      <reference reference-type="hard">
        <reference-target target-type="application" provider-name="sap.com">saml2_lib</reference-target>
      </reference>
      ```
2. Develop your J2EE Web application of the Web module project.
   a. Reference the interface for the external adapter
      *(com.sap.security.saml2.lib.interfaces.extensions.SAML2IdPExternalAdapter)* or for the attribute provider *(com.sap.security.saml2.lib.interfaces.extensions.SAML2ExternalAttributesProvider)* from your project by including the JAR files as libraries in the project’s Java build path.

      You have to add the JAR files from the following folder: `\<SAP AS Java host>\sapmnt\SID\instance_number\j2ee\cluster\apps\sap.com\saml2_lib\app_libraries_container`

   b. Implement the interface by developing your own logic for the extension in the script.
   c. Build an EAR file
   d. Deploy the EAR file.

**Related Information**

- Creating Enterprise Application Projects
- Editing the application-j2ee-engine.xml
- Tutorial for Developing Identity Provider Extensions [page 58]
- Deploying and Undeploying with SAP NetWeaver Developer Studio

**1.4.4.1.1 Tutorial for Developing Identity Provider Extensions**

The identity provider (IdP) functionality can be enhanced with custom implemented plug-ins. Each plug-in has to implement specific Java interface and be registered in the IdP at runtime. All IdP extension interfaces define `getName() : String` and `getDescription() : String` methods that should be implemented. The returned name has to be unique for the system and the description is optional. The administrator can configure which of the registered extensions needs to be invoked by the identity provider. The name and the description returned by the extension are visible in the SAP NetWeaver Administrator as well as the application that has registered this particular extension.

In addition to the implementation of the extension interface, you also have to implement a context listener that is applicable to both IdP extensions.

**1.4.4.1.1.1 Develop an Extension for External Adapter**

You use this extension to influence the authentication and to prevent issuing of specific SAML 2.0 assertions. To do this, you have to implement the `com.sap.security.saml2.lib.interfaces.extensions.SAML2IdPExternalAdapter` interface. The following two methods are invoked by the identity provider:
- **isAssertionIssuingDenied(assertionDetails : SAML2AssertionDetails) : boolean**
  The identity provider invokes this method each time before it issues a SAML 2.0 assertion. The custom implementation receives details regarding the assertion to be issued and can return true if the assertion should not be issued. The received details include the logon ID of the authenticated user, the name of the recipient service provider, the current HttpServletRequest and others.

- **applyAuthenticationRequirements(details : SAML2IdPAuthenticationRequestDetails, requirements : SAML2IdPAuthenticationRequirements) : SequenceControl**
  This method is invoked by the identity provider each time an authentication request arrives. The custom implementation receives details regarding the authentication request and can decide whether to request specific authentication and to force authentication.

---

**Note**

**Example of External Adapter Class**

```java
package com.sap.example;
import java.util.Collections;
import com.sap.security.saml2.lib.extensions.SAML2ExtensionService;
import com.sap.security.saml2.lib.interfaces.extensions.SAML2AssertionDetails;
import com.sap.security.saml2.lib.interfaces.extensions.SAML2IdPAuthenticationRequestDetails;
import com.sap.security.saml2.lib.interfaces.extensions.SAML2IdPAuthenticationRequirements;
import com.sap.security.saml2.lib.interfaces.extensions.SAML2IdPExternalAdapter;
import com.sap.security.saml2.lib.interfaces.extensions.SequenceControl;

/**
 * Represents an external adapter that can control the identity provider behavior.
 * This supplier could be registered and unregistered using
 * SAML2ExtensionService#registerIdPExternalAdapter(SAML2IdPExternalAdapter) and
 * SAML2ExtensionService#unregisterIdPExternalAdapter(SAML2IdPExternalAdapter) methods.
 * @see SAML2ExtensionService
 */
public class ExampleCustomIdPAdapter implements SAML2IdPExternalAdapter {
    /**
     * Returns the name of the plug-in that will be treated as an identifier of this external plug-in.
     * It must be unique per implementation of a plug-in.
     * @return Name of the external plug-in. Must not return null.
     */
    @Override
    public String getName() {
        return "ExampleIdPAdapter";
    }
    /**
     * Returns an optional short description of the plug-in.
     * @return Description of the external plug-in or null if the plug-in does not provide description
     */
    @Override
    public String getDescription() {
        return "ExampleIdPAdapter description";
    }
}
```
return "External adapter that request force authentication using username and password " +
"authentication. Also it denies the issuing of SAML 2.0 assertions to a specific " +
"service provider for user 'Administrator';
}
/**
 * Provides a way to control the authentication that the identity provider
 * is to perform.
 * 
 * The registered adapters are invoked one after another in order that is
 * configured by the
 * administrator of the identity provider (IdP). The adapter can control
 * whether the identity provider
 * to continue with the invocation of the next adapter (e.g. in case this
 * adapter does not have
 * specific requirements and let the next adapters to be able to request
 * such) or to stop the
 * invocation sequence and to fulfill the currently set (if any)
 * authentication requirements.
 * This is controlled by the returned {link SequenceControl}.
 * 
 * Note that the invocation of this method does not necessarily mean that
 * the identity provider
 * is to perform authentication. It will decide whether authentication is
 * needed and what
 * authentication to trigger based on the current IdP session, previous
 * authentications in the
 * same session, the requirements from the external service provider and
 * the result from this
 * method call.
 * 
 * But any authentication triggered by the identity provider will be
 * preceded by invocation of
 * this method.
 * 
 * @param details
 * Details for the current authentication requested from the
 * identity provider
 * 
 * @param requirements
 * The authentication requirements, which can be changed by the
 * external adapter in order
 * to request them to be fulfilled by the identity provider.
 * 
 * @return Whether to continue with the next adapters and give them a
 * possibility to provide
 * their authentication requirements or to stop invoking other
 * adapters and request
 * the identity provider to fulfill the currently set
 * authentication requirements.
 * 
 * If the adapter returns null, the behavior is nondeterministic.
 */
@Override
public SequenceControl
applyAuthenticationRequirements(SAML2IdPAuthenticationRequestDetails details,
SAML2IdPAuthenticationRequirements requirements) {
if (details.getTrustedSPName().equals("some.trusted.service.provider")) {
requirements.setForceAuthentication(true);
String suthnContext = "urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport";
requirements.setAuthenticationContextClassRefs(Collections.singletonList(suthnContext));
}
// Instruct the IdP not to invoke the next configured adapters (if any)
return SequenceControl.STOP;
}
// Allow next configured extensions to request specific authentication
return SequenceControl.CONTINUE;
/**
 * Provides a way to control whether the identity provider to continue
 * issuing the SAML 2.0
 * assertion represented by the given assertionDetails or not.
 * @param assertionDetails
 * Details for the SAML 2.0 assertion that the identity provider will issue.
 * @return true if the assertion should not be issued or false
 * otherwise
 */
@Override
public boolean isAssertionIssuingDenied(SAML2AssertionDetails assertionDetails) {
if (assertionDetails.getRecipientSPName().equals("some.trusted.service.provider")
&&
assertionDetails.getAuthenticatedUserLogonId().equalsIgnoreCase("administrator")) {
    // SAML 2.0 assertion should not be issued for trusted service provider
    // with name
    // "some.trusted.service.provider" for user "Administrator"
    return true;
} else {
    // for any other cases the SAML 2.0 assertion can be issued
    return false;
}
}

1.4.4.1.1.2 Develop an Extension for an Attribute Provider

You use this extension to provide your own SAML 2.0 attributes to the SAML 2.0 assertions issued by the identity provider. To do this, you have to implement the com.sap.security.saml2.lib.interfaces.extensions.SAML2ExternalAttributesProvider interface. Its getAttribute(assertionDetails : SAML2AssertionDetails) : Collection<SAML2ExternalAttribute> method is invoked by the identity provider each time before the provider issues the SAML 2.0 assertion. The custom implementation receives details regarding the assertion to be issued and can return zero or more SAML 2.0 attributes for this assertion. The received details include the logon ID of the authenticated user, the name of the recipient service provider, the current HttpServletRequest and others.

Note
Example of IdP Attribute Provider Class

package com.sap.example;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;
import java.util.Collections;
import java.util.List;
import java.util.Map;
import javax.xml.namespace.QName;
import com.sap.security.saml2.lib.extensions.SAML2ExtensionService;
import com.sap.security.saml2.lib.interfaces.extensions.SAML2AssertionDetails;
import com.sap.security.saml2.lib.interfaces.extensions.SAML2ExternalAttribute;
import com.sap.security.saml2.lib.interfaces.extensions.SAML2ExternalAttributesProvider;

/**
 * Represents an external provider of SAML 2.0 Attributes for issued SAML 2.0 Assertions.
 * This provider could be registered and unregistered using
 * @link SAML2ExtensionService#registerExternalAttributesProvider(SAML2ExternalAttributesProvider)
 * and
 * @link SAML2ExtensionService#unregisterExternalAttributesProvider(SAML2ExternalAttributesProvider)
 * methods.
 * @see SAML2ExtensionService
 */
public class ExampleCustomAttributesProvider implements SAML2ExternalAttributesProvider {
    /**
     * Returns the name of the plug-in that will be treated as an identifier of this external
     * plug-in.
     * It must be unique per implementation of a plug-in.
     * @return Name of the external plug-in. Must not return <code>null</code>.
     */
    @Override
    public String getName() {
        return "ExampleAttributeProvider";
    }
    /**
     * Returns an optional short description of the plug-in.
     * @return Description of the external plug-in or <code>null</code> if the plug-in does not
     * provide description
     */
    @Override
    public String getDescription() {
        return "Attribute provider that add the IP address of the client " +
                "and a multi-valued attribute to each assertion. Logon ID of " +
                "the user is added to assertions issued for a specific service provider";
    }
    /**
     * Returns SAML 2.0 attributes that need to be added for the currently
     * issued SAML 2.0 assertion.
     * The assertion is denoted with the given details parameter.
     * @param details
     * @return Collection SAML 2.0 attributes or <code>null</code> if this supplier
decides not to provide any attributes for the currently issued SAML 2.0 assertion

```java
@Override
public Collection<SAML2ExternalAttribute> getAttributes(SAML2AssertionDetails details) {
    Collection<SAML2ExternalAttribute> result = new ArrayList<SAML2ExternalAttribute>();

    result.add(new SAML2ExternalAttributeImpl("ip.address", details.getHttpRequest()));
    // multi-value attribute
    List<String> attributeValues = Arrays.asList(new String[]{"Engineers", "Purchasers" });
    result.add(new SAML2ExternalAttributeImpl("member.of", attributeValues));
    if (details.getRecipientSPName().equals("some.trusted.service.provider")) {
        // this attribute "logonId" will be included only in Assertions issued for Service Provider
        // with name "some.trusted.service.provider"
        result.add(new SAML2ExternalAttributeImpl("logonId", details.getAuthenticatedUserLogonId()));
    }
    return result;
}
```

```java
private static class SAML2ExternalAttributeImpl implements SAML2ExternalAttribute {
    private String name;
    private List<Object> values;

    public SAML2ExternalAttributeImpl(String name, Object value) {
        this(name, Collections.singletonList(value));
    }

    public SAML2ExternalAttributeImpl(String name, List<Object> values) {
        this.name = name;
        this.values = new ArrayList<Object>(values);
    }

    @Override
    public String getName() {
        return name;
    }

    @Override
    public List<Object> getAttributeValues() {
        return values;
    }

    @Override
    public String getFriendlyName() {
        return null;
    }

    @Override
    public String getNameFormat() {
        return null;
    }

    @Override
    public Map<QName, String> getOtherXMLAttributes() {
        return null;
    }
}
```
1.4.4.1.3 Develop a Context Listener

```java
package com.sap.example;
import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;
import com.sap.security.saml2.lib.common.SAML2Exception;
import com.sap.security.saml2.lib.extensions.SAML2ExtensionService;
import com.sap.tc.logging.Location;
import com.sap.tc.logging.Severity;
public class ContextListener implements ServletContextListener {
    private ExampleCustomIdPAdapter adapter;
    private ExampleCustomAttributesProvider provider;
    private Location location = Location.getLocation(ContextListener.class);
    @Override
    public void contextDestroyed(ServletContextEvent arg0) {
        if (adapter != null) {
            SAML2ExtensionService.getInstance().unregisterIdPExternalAdapter(adapter);
        }
        if (provider != null) {
            SAML2ExtensionService.getInstance().unregisterExternalAttributesProvider(provider);
        }
    }
    @Override
    public void contextInitialized(ServletContextEvent arg0) {
        try {
            adapter = new ExampleCustomIdPAdapter();
            SAML2ExtensionService.getInstance().registerIdPExternalAdapter(adapter);
            provider = new ExampleCustomAttributesProvider();
            SAML2ExtensionService.getInstance().registerExternalAttributesProvider(provider);
        } catch (SAML2Exception e) {
            location.traceThrowableT(Severity.ERROR, "registration problem", e);
        }
    }
}
```

1.4.4.2 Configure SAP NetWeaver Administrator

Prerequisites

The host AS Java must be compliant with the required versions for using IdP extensions. For more information, see System Requirements [page 23].
Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose Configuration ➤ Authentication and Single Sign-On ➤ SAML 2.0 ➤ Local Provider ➤.
3. Choose the Extensions tab.
4. Choose the Edit button.
5. Select Attribute Providers extension type and add or remove the attribute providers.
6. Select External Adapters extension type and add or remove the adapters.

**Note**
The order of the external adapters matters because the identity provider processes them sequentially.

7. Save your configuration.

Related Information

Implement an Identity Provider Extension [page 57]

1.4.5 Configuring Policy Scripts for Identity Provider Extensions

You can set policy scripts for identity provider extensions by specifying them in the user interface.

Context

This section explains how to configure policy scripts for an external adapter and attribute provider. For an external adapter, you have to implement authentication and assertion policies. For an attribute provider, you have to implement a script for policy-based assertion attributes. You configure these policies either for a specified trusted service provider or for all trusted service providers. Before implementing the policy script, you have to set the extensions. To do this, proceed as follows:

**Note**
If you have used SAP Single Sign-On version 2.0 SP06 or lower, and you have custom policy scripts, when you open the Trusted Providers link on the SAML 2.0 tab of the SAP NetWeaver Administrator UI you will be prompted to migrate the policy scripts to the new Policy Script Administration Console. You have the following options:

- You can ignore the migration and continue to work with the policy scripts as in version SAP Single Sign-On 2.0. For more information, see Access Policies Implementation Guide.
You can migrate the existing policy scripts to the Policy Script Administration Console, where you can create and manage new policy script versions. For more information, see Policy Scripts Implementation Guide.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose the Configuration tab.
3. Choose Authentication and Single Sign-On ➤ SAML 2.0 ➤ Local Provider ➤ Extensions ➤.
4. Choose the Edit button.
5. Choose the External Adapter or Attribute Provider extension from the Extension Type dropdown box.
6. Choose the Add button to see all available extensions and add the policy extension that has the following source application: sap.com/access.policies.idp.integration.
   ○ For an external adapter policy, add AccessPolicyAdapter.
   ○ For an attribute provider policy, add AccessPolicyAttributesProvider.
7. To remove a selected extension, choose the Remove button.
8. Save your configuration.

Next Steps

- Implement a Policy Script for an External Adapter [page 66]
- Implement a Policy Script for an Attribute Provider [page 68]

1.4.5.1 Implement a Policy Script for an External Adapter

Prerequisites

You have set the AccessPolicyAdapter extension. For more information, see Configuring Policy Scripts for Identity Provider Extensions [page 65].

Context

By implementing policy scripts for an external adapter, you set conditions for the authentication process of the identity provider. For this configuration, you specify an authentication policy, which influences the authentication performed by the identity provider, and an assertion policy, which decides whether or not the assertion is issued.
You can configure these policy scripts either for a specified trusted service provider or for all trusted service providers.

### 1.4.5.1.1 Implementation for a Trusted Service Provider

**Procedure**

1. Implement the policy script for a trusted service provider.
   
   You have to create a policy script of type *Procedure* in the *Policy Script Administration Console*, which is accessed with the quick link `/ssoadmin/scripts`. For more information, see *Working with Policy Scripts*.

2. Start SAP NetWeaver Administrator with the quick link `/nwa/auth`.

3. Choose the *Configuration* tab.

4. Choose *Authentication and Single Sign-On* > *SAML 2.0* > *Trusted Providers*.

5. Select the service provider that you want to configure the policy script for.

6. Choose the *Authentication Requirements* tab.

7. Choose the *Edit* button.

8. From the dropdown lists in the *Authentication* and *Assertion Policy Scripts* sections, choose the corresponding authentication and assertion policy scripts.

9. Save your configuration.

### 1.4.5.1.2 Implementation for All Trusted Service Providers

**Procedure**

1. Implement the authentication and assertion policies for all trusted service providers.

   You have to create policies of type *Procedure* in the *Policy Script Administration Console*, which is accessed with the quick link `/ssoadmin/scripts`. For more information, see *Working with Policy Scripts*.

2. Start SAP NetWeaver Administrator with the quick link `/nwa/auth`.

3. Choose the *Configuration* tab.

4. Choose *Infrastructure* > *Java System Properties*.

5. Choose the *Applications* tab.

6. Select *access.policies.idp.integration* application.

7. Modify the value of the *idp.authentication.policy.name* property to be the name of the authentication policy that is created in the *Policy Script Administration Console*.

8. Modify the value of the *idp.assertion.denying.policy.name* property to be the name of the assertion policy that is created in the *Policy Script Administration Console*.

9. Save your configuration.
If the operation is successful, you receive the message Changes were saved successfully at the top of the page.

1.4.5.2 Implement a Policy Script for an Attribute Provider

Prerequisites

You have set the AccessPolicyAttributesProvider extension. For more information, see Configuring Policy Scripts for Identity Provider Extensions [page 65].

Context

By implementing a policy script for an attribute provider, you define the attributes passed to the issued SAML 2.0 assertion.

You can configure this policy script either for a specific trusted service provider or for all trusted service providers.

1.4.5.2.1 Implementation for a Trusted Service Provider

Context

For this implementation, you configure a policy script for a name ID format of the service provider. For more information about the name ID formats for a trusted service provider, see Identity Federation.

Procedure

1. Implement the policy script for a trusted service provider.
   In the Policy Script Administration Console, accessed with the quick link /ssadmin/scripts, create a policy script of type Procedure that provides SAML 2.0 attributes. For more information, see Working with Policy Scripts.
2. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
3. Choose the Configuration tab.
5. Select the service provider that you want to configure the policy script for.
6. Choose the Identity Federation tab
7. Choose the *Edit* button and select the *NameID* format.
8. Under *Details of NameID Format* "<name ID format>"*, choose the *Policy-Based Assertion Attributes* tab.
9. From the dropdown list choose the corresponding policy script created in the *Policy Script Administration Console*.
10. Save your configuration.

**1.4.5.2 Implementation for All Trusted Service Providers**

**Procedure**

1. Implement a script for all trusted service providers.

   You have to create a policy of type *Procedure* in the *Policy Script Administration Console*, which is accessed with the quick link /ssoadmin/scripts. For more information, see *Working with Policy Scripts*.
2. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
3. Choose the *Configuration* tab.
4. Choose *Infrastructure > Java System Properties*.
5. Choose the *Applications* tab.
6. Select the *access.policies.idp.integration* application.
7. Modify the value of the *idp.attributes.provider.policy.name* property to be the attribute provider policy that is created in the *Policy Script Administration Console*.
8. Save your configuration.

   If the operation is successful, you receive the message *Changes were saved successfully at the top of the page.*

**1.4.6 Configuring the Metadata and Metadata Access**

Other SAML 2.0 providers can trust a SAML 2.0 provider of SAP NetWeaver Application Server (AS) Java by accessing the metadata XML file. You can configure how the AS Java offers the metadata: by URL or by download from the configuration user interface. You can also determine if the file should be signed digitally and whether it includes any contact information from your organization.

**Procedure**

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose *SAML 2.0 > Local Provider*.
3. Choose the *Metadata* tab.
4. Choose the *Edit* pushbutton.
5. Determine if you want to enable access to the metadata URL.
   Public access to this URL is disabled by default. By providing access to this URL, you enable your partners to easily configure their SAML 2.0 provider to trust your system. But you must consider the security risk of exposing the endpoints and bindings you support to potential attackers. To enable public access, enter Enabled in the Public Access field.

6. Determine if you want the metadata digitally signed.
   Many SAML 2.0 providers require metadata XML files to be signed digitally to guarantee that the file is genuine. Add a digital signature to ensure that other providers can verify that the metadata XML file came from your SAML 2.0 provider.
   To have the provider sign the metadata, enter Yes in the Sign Metadata field.

7. Enter any contact data.
   Adding human-readable contact data to the metadata XML file, enables the operators of other SAML 2.0 providers to contact you or your organization with questions about your provider and the services you offer.

8. Save your entries.

Related Information

Accessing the Metadata XML of SAML Identity Providers [page 90]

1.4.7 Configuring the Redirect Application

The redirect application is used to redirect the user to the specified redirect site by a GET method (the redirect site application does not preserve the POST method). The Redirect URL specifies the URL that the redirect application uses to access the Redirect Site. This URL can be in the form of a relative or absolute path. You can specify additional parameters to the redirect site application by adding them at the end of the URL address. If the parameters are added to the URL of the redirect link, they must be URL encoded. The redirect link is the URL link that the user chooses in the browser to access the redirect site.

Context

To enable and configure the use of the redirect application, you have to do the following:

Procedure

1. Go to the tab that sets the URL and parameters for the redirect application.
   To access the redirect application, choose Configuration Management ➤ Authentication and Single Sign-On ➤ SAML 2.0 ➤ Local Provider ➤ Proxying Settings.
2. Set the redirect URL with the necessary parameters.

The redirect application table allows you to set a number of redirect sites and their URLs. To make these settings, enter the name of the redirect site in the RedirectSite field and its URL in the Redirect URL field. In this way, you specify the name to which the URL maps.

3. Set the default redirect URL, if necessary.

By setting a default URL in the Default Redirect URL field, you specify which URL the system uses, if you have not set the parameters for the redirect site.

Note

Donna Moore has configured the system to use a redirect site portal with the URL http://portal.acme.com/irj. In addition, she decides to set two parameters, page with a value mainpage and my_name with a value Donna Moore. Therefore, Donna changes the address of the redirect site portal to http://portal.acme.com/irj?page=mainpage in the Redirect URL field. When Donna chooses the redirect link https://idp.company/saml2/idp/redirect?RedirectSite=portal%3Fmy_name%3DDonna+Moore, the application redirects her to the site with the URL http://portal.acme.com/irj?page=mainpage&my_name=Donna+Moore.

1.4.8 Configuring the Lifetime of Identity Provider Sessions

Your strategy for identity provider sessions depends on how often your users interact with the identity provider and how many users you have for your resources. Each HTTP session uses the Internet Communication Manager (ICM) and database resources. You must balance this with how often you can expect to force your users to re-authenticate.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 Local Provider.
3. Choose the Identity Provider Settings tab.
4. Under Single Sign-On Service, configure the lifetime of identity provider sessions in accordance with the table below.

Caution

If you set a value that is too high, your database fills with expired sessions causing a decrease in system performance or, in the worst case, causing your system to become unresponsive.
### Session Timeout

This property determines how long the identity provider keeps a session alive without modification. An identity provider modifies the session when a service provider requests re-authentication or authentication with another authentication context.

### Cleanup Interval for Expired Sessions

This property determines how often expired identity provider sessions are deleted from the database.

### Related Information

Monitoring Identity Provider Sessions [page 84]

### 1.4.9 Configuring the Validity Period for SAML Messages

To secure your service provider, limit the validity period of Security Assertion Markup Language (SAML) 2.0 messages that the service provider receives. However, delays in computer networks and skewed clocks can cause otherwise valid messages to become invalid. Use this procedure to configure how much time before and after the instant the identity provider created the SAML message that the service provider can accept it. The time of message creation is recorded in the SAML message in the `IssueInstant` attribute.

### Procedure

1. Start SAP NetWeaver Administrator with the quick link `/nwa/auth`.
2. Choose `SAML 2.0 > Local Provider`.
3. Choose the `Edit` pushbutton.
4. Choose the `General Settings` tab.
5. Under `Miscellaneous`, configure the validity period for SAML messages by entering how many minutes before and after a message was issued that it becomes invalid.

   **Note**
   
   The default validity period is up to 5 minutes before and no more than 10 minutes after the message was created.

6. Save your entries.
1.4.10 Disabling IdP-Initiated and SP-Initiated SSO and SLO

Under SAML, clients can initiate Single Sign-On (SSO) and Single Log-Out (SLO) at either the identity provider (IdP) or the service provider (SP). You can control whether the service provider accepts SAML messages initiated at the service provider or identity provider. Thus you determine what kind of access clients have to your SAML landscape.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 Local Provider.
3. Choose the Identity Provider Settings tab.
4. Under Single Sign-On Service, deselect the supported types of SSO you do not want to allow.
5. Under Single Log-Out Service, deselect the supported type of SLO you do not want to allow.
6. Save your entries.

1.4.11 Disabling the SAML 2.0 Provider

Use this procedure when you no longer need SAP NetWeaver Application Server (AS) Java to perform as a Security Assertion Markup Language (SAML) 2 identity provider.

Prerequisites

You have ensured that all resources protected by SAML 2.0 have an alternative authentication mechanism.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0.
3. Choose the Local Provider link.
4. Disable or delete the configuration.
   ○ To disable the configuration, choose the Disable pushbutton.
     The AS Java saves the SAML 2.0 configuration. You can re-enable SAML 2.0 at any time.
   ○ To delete the configuration, choose the Delete Configuration pushbutton.
     The AS Java deletes the SAML 2 configuration. You can decide to keep or delete the related keystore entries and certificates.
1.4.12 Enabling HTTP Access to SAML Endpoints

Some user agents cannot support Secure Socket Layer (SSL) to protect HTTP. To support such users, you can enable the access to SAML endpoints over HTTP.

Context

Recommendation

We recommend the use of SSL to prevent eavesdroppers from accessing authentication information in SAML messages. You can protect authentication information with encryption and by using the back channel.

For more information, see the following:
- Securing SAML Bindings
- Configuring Back-Channel Communication

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 ➔ Local Provider ➔
3. Choose the Edit pushbutton.
4. Choose the General Settings tab.
5. Under Miscellaneous, choose Yes in the Allow HTTP Access field.
6. Save your entries.

Results

If you change the configuration of your identity provider, update the configurations of any service providers that trust your identity provider to match.

Related Information

Securing SAML Bindings [page 80]
Configuring Front-Channel Communication [page 35]
1.4.13 Enabling Service Providers to Share Persistent Name IDs

A given persistent name ID exists only once per user pairing between an identity provider and a service provider. You can configure a group of service providers to share the persistent name IDs for each user. All service providers that take part in an affiliation, identify themselves to the identity provider with the affiliation ID. So long as the identity provider configuration shows the service provider is in that affiliation, the identity provider sends the persistent name ID configured for the user in the affiliation. The identity provider returns an error when a service provider makes an authentication request for an affiliation it is not a member of.

Prerequisites

You have configured your identity provider to use the persistent name ID format with the target service providers. For more information, see Configuring Identity Federation with Persistent Pseudonyms.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 Trusted Providers.
3. Show Affiliations.
4. Choose the Add pushbutton.
5. Enter an ID for the affiliation and a user attribute and namespace to store the persistent name ID. The identity provider writes the persistent name ID to the user record in this user attribute.
6. Add members to the affiliation.
   a. Select an affiliation and choose the Edit pushbutton.
   b. Under Members of affiliation "<affiliation_ID>", choose the Add pushbutton.
   c. Select one or more trusted service providers.
   d. Save your entries.
7. Configure any service providers that are to participate in the affiliation to use the same affiliation ID. For more information, see the documentation provided by your service provider vendor.

Related Information

Configuring Identity Federation with Persistent Pseudonyms [page 49]
1.4.14 Determining the Channel Used for SLO by the Identity Provider

When a service provider initiates Single Log-Out (SLO), the identity provider passes on SLO requests to other service providers using the channel configured for the default SLO endpoint for that provider. When the identity provider initiates SLO or if the local service provider of the identity provider initiates SLO, you can configure how the identity provider reacts.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose [SAML 2.0] [Local Provider].
3. Choose the Identity Provider Settings tab.
4. Under Single Log-Out Service, select the default channel for SLO.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back Channel</td>
<td>The identity provider passes on logout requests to service providers over a SOAP binding. If there is no SLO endpoint configured for a service provider, the identity provider does not attempt to send a logout request.</td>
</tr>
<tr>
<td>Front Channel</td>
<td>(Default) The identity provider passes on logout requests to service providers over the binding used by the default SLO endpoint for the service provider.</td>
</tr>
</tbody>
</table>

1.4.15 Including Legacy Systems in Your SAML 2.0 Landscape

Not all releases of SAP software support Security Assertion Markup Language (SAML) 2.0. These systems can still benefit from having SAML 2.0 in the system landscape. Configure your identity provider to issue log-on tickets when a user logs on. This enables legacy systems within the domain to perform Single Sign-On initiated by the SAML 2.0 authentication.

Prerequisites

- You have configured your legacy systems to accept log-on tickets.
- Ideally the user ID of users on the identity provider is identical to their user IDs on the legacy systems. If not, you must configure user mapping.
Context

i Note
If your identity provider is also a service provider, the log-on ticket is only issued once, even if the resource the user accesses is protected by the local identity provider and service provider.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 Local Provider.
3. Choose the Edit pushbutton.
4. Choose the General Settings tab.
5. Under Miscellaneous, enter on in the Legacy Systems Support (Issue Logon Ticket) field.
6. Save your entries.

1.4.16 Mapping Authentication Contexts to Log-in Modules

Prerequisites

- The authentication context you want to map to exists. All of the authentication contexts required by the standard system exist by default. You can add custom authentication contexts. For more information, see Adding Custom Authentication Contexts.
- A matching log-in module for the authentication context exists on the SAP NetWeaver Application Server (AS) Java. For more information about developing custom log-in modules, see the online documentation at http://help.sap.com.

Context

By default, the identity provider supports the following authentication contexts through the log-in modules mapped in the table below.

Mapping of SAML 2 Authentication Contexts to Log-in Modules

<table>
<thead>
<tr>
<th>Authentication Context</th>
<th>Login Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kerberos</td>
<td>SPNegoLoginModule</td>
</tr>
<tr>
<td>Password</td>
<td>BasicPasswordLoginModule</td>
</tr>
</tbody>
</table>
By changing these mappings or by adding new mappings you can control how users are authenticated at the identity provider for the authentication context requested by the service provider. You can also configure the properties necessary for the authentication at the identity provider in a mapping of this type. A configuration of this type is also helpful when you are configuring mappings to a customized log-in module. You define these properties in the settings for policy configuration, and they are configured for the authentication context at the identity provider. You can configure the policy configuration properties by choosing [Authentication Management ➤ Authentication and Single Sign-On ➤ Authentication ➤ Components ➤ Properties].

You also have to configure the log-in module(s) that the authentication context should map to. You can add log-in modules to the authentication stack by choosing [Configuration Management ➤ Authentication and Single Sign-On ➤ SAML 2.0 ➤ Local Provider ➤ Identity Provider Settings]. You also have to configure the options for the log-in module(s) by choosing [Authentication ➤ Login Modules ➤ Login Module Options].

When the identity provider receives the authentication request with the authentication context from the service provider, it checks that it supports the context and finds the log-in module to which it maps its authentication context. The authentication context uses the properties from the policy configuration for the identity provider authentication. If you have set a configuration for a third-party log-in module, you have to configure the authentication context(s) that map to the log-in module.

Example

The identity provider receives a Third-Party Custom authentication context and confirms that it supports the context because the authentication context is configured as supported in its database. This context has a mapping to a ThirdPartyLoginModule. The user sets a name, sap.com/saml2_idp*saml2_idp, for the policy configuration, and the identity provider stores the ThirdPartyLoginModule in its dynamic stack. Consequently, the identity provider uses the configured properties of the policy configuration and the authentication context’s mapping to the log-in module to complete the authentication.

Procedure

1. Start the SAML 2.0 configuration application.
2. Choose the Identity Provider Settings tab.
3. Choose the Edit pushbutton.
4. Choose the Add pushbutton.
5. Enter data as required.
6. Make any required configurations for HTTPS.
   ○ To enable Secure Sockets Layer support for the authentication context, select the HTTPS checkbox.
   ○ To make the authentication a default HTTPS authentication context, choose Copy to Default HTTPS Authentication Context and determine the order in which the identity provider should take the authentication contexts.
1.4.17 Principles and Configuration of the Scoping Element

The scoping element is used for identity provider proxy scenarios. In scenarios of this type, the authentication request contains this scoping element, which includes a list of the possible authenticating identity providers. Some identity providers do not support the scoping element, but it is possible to use them as authenticating identity providers in proxy scenarios. In that case, the scoping element is not sent with the authentication request.

Prerequisites

To configure the scoping element, make sure that all of the proxy settings are defined.

For more information about the proxy configurations, see the following:

- Identity Provider Proxy
- Configuring Identity Providers as Proxies

Context

i Note
If you expect the trusted identity provider to be a proxy identity provider, ensure that this setting is configured to "Yes". If you know for sure that the trusted identity provider acts only as an authenticating identity provider and does not support the scoping element, configure this setting to "No".

Procedure

2. Under the list of trusted providers, show Identity Providers.
3. Choose the Authentication Requirements tab.
4. Choose the Edit pushbutton.
5. Under the Proxying Settings section, select whether you want to use the scoping element. The Send Scoping Element option is set to Yes by default.

6. Save your selection.

**Related Information**

Identity Provider Proxy [page 16]
Configuring Identity Providers as Proxies [page 55]

**1.4.18 Securing SAML Bindings**

Depending on how you have configured the trust between your SAML identity provider and its trusted service provider, the SAML messages exchanged can include authentication-relevant or personal information. Information of this kind includes user ID, name, last name, address, and telephone number. Exposing such information may expose your network to risk from eavesdroppers or violate local compliance regulations.

**Prerequisites**

This procedure assumes you have already configured trust with a service provider and you want to change these settings as a follow on procedure. You can make these same settings during trust configuration. For more information, see Adding Service Providers.

→ **Recommendation**

This procedure also assumes that the public-key certificates you use to encrypt and check digital signatures have been configured. Although you can change the configuration public-key certificate with this procedure, we recommend that you perform an update of the trusted provider instead. For more information, see Updating the Configuration of Trusted Providers.

**Context**

The SAML standard provides signature and encryption configurations to protect SAML bindings:

- Digital signatures validate the service of the provider.
  - You configure what messages the identity provider signs and what messages must be signed by the service provider. The identity provider rejects unsigned messages that require signatures.
  - The identity provider supports the following digest algorithms for signing the messages:
    - SHA-1
    - SHA-256
Encryption makes sensitive information unreadable without decoding. You configure what information the identity provider encodes and what information must be encoded by the service provider. The identity provider rejects messages with unencrypted information, where encrypted information is required.

If your network configuration allows it, you can also use back-channel communication to protect the client from sensitive information. Even back-channel communication can require protection, if the communication directly between service provider and identity provider is not secure. For more information, see Configuring Back-Channel Communication.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 Trusted Providers.
3. Select a service provider and choose the Edit pushbutton.
4. Choose the Signature and Encryption tab.
5. If necessary, update the certificates for signing and encryption and the digest algorithm for signing the outgoing SAML 2.0 messages.

⚠️ Caution
For this configuration to work, the service provider has to support the same signing digest algorithm.

6. Choose the signature and encryption options for requests, responses, and assertions for Single Sign-On (SSO), Single Log-Out (SLO), and artifact resolution. The signature and encryption options must match with those of the service provider. If the service provider requires SAML assertions always be digitally signed and the identity provider never signs them, then the SAML configuration cannot function.

→ Recommendation
Give some thought to your encryption and signature options and make choices that make sense for your configuration. These also depend on the environment in which your SAML network is working. Systems that operate in a secured area behind a firewall have different requirements from systems exposed to the Internet. We have the following recommendations:

○ Encryption
Encrypt or require encryption for those elements that can expose authentication or other personal data about the users. If you use the transient or persistent name ID formats, these name IDs are already opaque. There is no need to encrypt these name IDs. The e-mail name ID format, however, can reveal the users real name and contact information. When using the transient and persistent name ID formats, you can send attributes. These attributes can also reveal personal information, which you should encrypt.

○ Digital signatures
The SAML standard provides many points in the process at which you can sign and check for signatures. Do this only where it makes sense. For example, you can require signature of the SAML assertion and the SAML response. It does not make sense for the identity provider to sign the SAML response and then pack it in a SAML assertion and sign it again before sending the assertion to the service provider. This would only make sense if you developed a custom process to separate...
the SAML response from the SAML assertion and send the response over a third party before the response is processed. You can further complicate the process by using the HTTP artifact binding and requiring signature of the artifact response. The identity provider signs the message three times.

7. Save your entries.
8. Ensure the configuration of the service provider matches the changes you made.

For more information, see the documentation supplied by the service provider vendor.

Related Information

Configuring Back-Channel Communication [page 31]
Adding Service Providers [page 39]
Updating the Configuration of Trusted Providers [page 42]

1.4.19 Setting the Timeout for Database Lock in Clusters

When the identity provider modifies tables in the database, it acquires a lock from the enqueue server to ensure instances in the cluster remain consistent. You can configure the length of time the identity provider waits for a lock. If no lock is returned, the identity provider returns an error.

Context

Under heavy traffic the identity provider returns more errors. Increase the timeout to prevent authorization requests from failing because of requests that time out while trying to acquire a lock. You can decrease the timeout period to prevent authorization requests from taking too long before coming back with an answer. However, there is a real danger that the request simply fails when the cluster is too busy.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 Local Provider.
3. Choose the Identity Provider Settings tab.
4. Choose the Edit pushbutton.
5. Under Miscellaneous, enter a value in the Lock Obtaining Timeout field.
6. Save your entries.
1.5 Operations and Monitoring

This section describes activities you can perform with the identity provider to manage name IDs, monitor SAML sessions, to direct user agents with URL parameters, and to access the metadata of the SAML 2.0 provider.

1.5.1 Managing Name IDs

The name ID is the common identifier between the SAML 2.0 identity provider and the service provider. By setting the name ID for a user on SAP NetWeaver Application Server (AS) to the same name ID for a user on a service provider, you federate the two accounts. By removing the name ID for a user, you defederate the accounts.

Context

Use this procedure to federate and defederate accounts or to identify the name ID used by a user account for different service providers.

Procedure

1. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
2. Choose SAML 2.0 > Name ID Management.
3. Enter a user and choose a name ID format.
   To select multiple users, select a role or group.
   
   **Note**
   
   You can only generate random persistent name ID or remove persistent name IDs when you select multiple users with a role or group.

4. Enter data as required.
   ○ Federate single user accounts by editing the name ID of the user.
   ○ Defederate one or more user accounts by removing the name ID.
   The source for the name ID format determines if you can edit the name ID. For some sources, you can only view the name ID. The table below lists, which name ID sources for the name ID formats are editable.
### Editable and Read-Only Sources for Name IDs per Name ID Format

<table>
<thead>
<tr>
<th>Name ID Format</th>
<th>Editable Sources</th>
<th>Read-Only Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Kerberos</strong></td>
<td>User attribute</td>
<td>○ Active Directory Server (ADS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ JAAS Subject</td>
</tr>
<tr>
<td><strong>Persistent</strong></td>
<td>User attribute</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i Note&lt;br&gt;Generate random or remove only</td>
</tr>
<tr>
<td><strong>Unspecified</strong></td>
<td>User attribute</td>
<td>○ Logon alias</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ Logon ID</td>
</tr>
<tr>
<td><strong>Windows Name</strong></td>
<td>User attribute</td>
<td>Active Directory Server (ADS)</td>
</tr>
<tr>
<td><strong>X.509 Subject Name</strong></td>
<td>User attribute</td>
<td>JAAS Subject</td>
</tr>
</tbody>
</table>

5. Save your entries.

### 1.5.2 Monitoring Identity Provider Sessions

When a user authenticates at the identity provider with SAML, the identity provider creates an identity provider session. The identity provider session tracks the trusted service providers where the user has sessions authenticated with SAML.

**Prerequisites**

You must have a user assigned to a role with the required authorizations. For more information, see *Authorizations*.

**Context**

Use this procedure to see which users are authenticated at the identity provider and what sessions they have on service providers. You can also terminate sessions to remove abandoned session from your network. Problems in the network or with the service providers themselves can create abandoned sessions.

To force log-out from the user agent, direct the agent to the Single Log-Out (SLO) URL of the identity provider. For more information, see *Performing Identity Provider Initiated Single Log-Out*. 
Procedure

1. Start the identity provider sessions application.
   Enter the following URL:
   \[\text{http(s)}://<\text{hostname}>:<\text{port}>/saml2/idp/sessions.\]
2. Search for users.
   You can refine the search with the IP address of the user agent or status of the identity provider session.
3. View the search results.
   - For each user authenticated with SAML, you can view the list of identity provider sessions that user has. A user can have multiple identity provider session, if they log on with different user agents, such as in different browser windows.
   - For each identity provider session, you can view the list of session participants, the user has with each security provider as well as the name ID and format used for that session.
4. Terminate any unwanted identity provider sessions.

Related Information

- Authorizations [page 24]
- Performing Identity Provider-Initiated Single Log-Out [page 89]

1.5.3 Auditing and Performance Monitoring

The identity provider uses the audit logs and performance traces of the host SAP NetWeaver Application Server (AS) Java to record audit and performance information.

Audit Information

The identity provider uses the log viewer of the AS Java under the /System/Security/Authentication category.

The identity provider records the following information:

- Log-out responses
- Received log-out responses
- Authentication requests

Performance Information

The identity provider uses the Java system reports of the AS Java under the /Performance/IDP category.
The identity provider records the following information from the artifact response service:

- Successful calls
- Failed calls
- Response time

1.5.4 Troubleshooting on the Identity Provider

Context

Use the security troubleshooting wizard to gather traces relating to authentication problems with SAML 2.0.

Procedure

1. Start SAP NetWeaver Administrator.
2. Choose \(\text{Problem Management} \rightarrow \text{Logs and Traces} \rightarrow \text{Security Troubleshooting Wizard}\).
3. Enter \(\text{SAML 2.0 (Info)}\) as the incident type for diagnostics.
4. Choose the \(\text{Start Diagnostics}\) pushbutton.
5. Reproduce the problem.
6. Choose the \(\text{Stop Diagnostics}\) pushbutton.
7. Review the traces.
   - Try collecting traces on the service provider as well, in case that is the source of the problem. If you still require help, collect traces again with the incident type \(\text{SAML 2.0 (Debug)}\) and provide them to SAP Support.

1.5.5 Performing Identity Provider-Initiated Single Sign-On

Use this procedure to start the Single Sign-On (SSO) process at the identity provider, instead of starting with the user agent at the service provider. Possible reasons for doing this include:

Context

- To ensure the user agent has visited the identity provider before visiting the service provider. By visiting the identity provider first, you ensure that the identity provider has had an opportunity to note itself in the common domain cookie, enabling the service provider to discover the identity provider on a subsequent visit.
- To reduce the number of round-trips in your landscape.
Starting at the service provider always redirects the user agent to the identity provider. By starting at the identity provider, you save at least one round-trip.

- To make your identity provider the single point of access.
  Perhaps your portal is the host of your identity provider. Since all users start here anyway, you do not have to send them to the service providers and then back to the portal before sending them to the service provider.
- To start an identity provider-initiated SSO in an identity provider proxy scenario.
  For more information, see the topic "Identity Provider Proxy" in the identity management documentation.

You can also use redirect application options for configuring where the identity provider can redirect the user.

This procedure merely outlines the steps you need to get a user agent to perform. The details of implementation depend on your scenario.

**Procedure**

1. Direct the user agent to your SSO endpoint.
   The default endpoint is: `saml2/idp/sso`.

2. Add any URL parameters required for your scenario.
   The following table lists the URL parameters you can use for identity provider-initiated SSO.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>saml2sp</td>
<td>Yes</td>
<td>Name of the SAML 2 service provider for which SSO is performed.</td>
</tr>
<tr>
<td>RelayState</td>
<td>No</td>
<td>Relay state forwarded to the service provider with the SAML response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For SAP service providers, you can map this value to a resource of the service provider.</td>
</tr>
<tr>
<td>saml2endpointindex</td>
<td>No</td>
<td>Enter the index number 1-9 of the endpoint of the assertion consumer service of the service provider as the target of the SAML response. Otherwise the identity provider uses the default endpoint configured for the trusted service provider.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A non-digit value or a value for an index entry that is not configured returns an error message.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Mandatory</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>saml2binding</td>
<td>No</td>
<td>Ignored if saml2enpointindex is specified. Choose one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ post</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requires the identity provider to use the HTTP POST binding.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ artifact</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requires the identity provider to use the HTTP artifact binding.</td>
</tr>
<tr>
<td>saml2nameidformat</td>
<td>No</td>
<td>Name ID format for the authentication request. The service provider must be configured to support this format in the configuration of the trusted service providers of the identity provider.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The values for the parameter are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ email</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ kerberos</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ persistent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ transient</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ unspecified</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ windowsname</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ x509</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name of the affiliation for which SSO is performed. The identity provider returns an error if the affiliation does not exist or the requested service provider is not a member of the affiliation. The identity provider ignores this parameter if the name ID format is not persistent.</td>
</tr>
</tbody>
</table>

Example

https://idp.example.com:50001/saml2/idp/sso?sam12sp=sp.example.com
https://idp.example.com:50001/saml2/idp/sso?sam12sp=sp.example.com&RelayState=portal

3. Configure a redirect site, if necessary.
You use a redirect application to allow the system to redirect you to another site. The system redirects the user to the specified redirect site by a GET method (the redirect site application does not preserve the POST method). In addition, you can set parameters to the redirect URL address. For more information, see Configuring the Redirect Application.
Results

How the system reacts depends on whether the user has a security session on the identity provider already or not.

- **A session exists**
  The identity provider generates a response referring to the existing session and redirects the user agent according to the configuration of the URL parameters.

- **No session exists**
  The identity provider triggers authentication. If successful, the identity provider generates a response and redirects the user agent according to the configuration of the URL parameters.

Related Information

Configuring the Redirect Application [page 70]

### 1.5.6 Performing Identity Provider-Initiated Single Log-Out

Use this procedure to start the Single Log-Out (SLO) process at the identity provider, instead of starting with the user agent at the service provider. One reason to do this might be to enable the user to initiate global log out at a central portal page.

Context

This procedure merely outlines the steps you need to get a user agent to perform. The details of implementation depend on your scenario.

Procedure

1. Direct the user agent to your SLO endpoint.
   The default endpoint is: `/saml2/idp/slo`.
2. Add any URL parameters required for your scenario.
   Use the URL parameter, `RelayState`, to provide a relative URL to which the identity provider redirects after the SLO process is complete.
Results

Using URLs, you can trigger logout in all security sessions on the service provider and the security session on the identity provider. After successful logout, the identity provider either displays a success message or redirects to the URL provided in the relay state.

Note

https://idp.example.com:50001/saml2/idp/slo

1.5.7 Accessing the Metadata XML of SAML Identity Providers

The easiest way to trust a SAML 2.0 identity provider is to import its metadata XML file.

Prerequisites

- The SAML identity provider is enabled.
- You have configured the endpoints for Single Sign-On (SSO), Single Log-Out (SLO), artifacts, and SOAP you want to support. Any endpoints you configure later require you to manually reconfigure your service provider or reimport the metadata XML file.
- You have configured the metadata XML file and access. The configuration of the XML file determines whether or not the metadata file is signed digitally. It also determines how the metadata file can be accessed, by URL or download from the configuration user interface. For more information, see Configuring the Metadata and Metadata Access.

Caution

The hostname and protocol generated for the identity provider endpoints in the metadata XML file are the same as the hostname and protocol you use to access the metadata XML file. Use the same hostname and protocol you expect the service provider to use, when it accesses the identity provider endpoints. If you use a hostname that the service provider cannot resolve, or a protocol that the service provider cannot use, connections from the service provider fail.

Context

The metadata XML file includes the following:
- Address and name of the identity provider
• List of endpoint configurations the identity provider supports
• Public-key certificates for decryption and checking of the identity provider’s digital signature

This procedure explains how to access the metadata XML file of the identity provider of the SAP NetWeaver Application Server (AS) Java.

Procedure

• Downloading the Metadata XML File
  a. Start SAP NetWeaver Administrator with the quick link /nwa/auth.
  b. Choose SAML 2.0 Local Provider.
  c. Choose the Download Metadata pushbutton.
     If you require the metadata to be signed, you have the option to select another public-key certificate to sign the metadata. Use this option if you already have another method of trust set up to sign the metadata instead of the provider certificate.
     To use this option clear the Use Provider Signing Keypair checkbox and select the required keystore entry.
  d. Choose Download Metadata and save the XML file.

• Accessing the URL of the Metadata XML File

When configuring the service providers you want your SAML identity provider to trust, enter the following URL for the AS Java host system: https://<hostname>:<port>/saml2/metadata.

i Note
To access the metadata XML file with HTTP, you must enable HTTP access to the SAML service provider. For more information, see Enabling HTTP Access to SAML Endpoints.

Related Information

Configuring the Metadata and Metadata Access [page 69]
Enabling HTTP Access to SAML Endpoints [page 74]
Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP’s gross negligence or willful misconduct.
- Links with the icon: You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP’s gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.