LearningZen

Single Sign-On (SSO) Specifications

LEARNINGZEN.COM

May, 2021

Contents

Acknowledgements

Documentation Development Team

LearningZen Support Team

Copyright Information

Trademarks

Disclaimer

Contact Information

LearningZen Gold Systems, Inc.
2121 S McClelland St #204
Salt Lake City, UT 84106

(877) 850-1214
support@learningzen.com

Overview

LearningZen supports Single Sign-On (SSO) SAML2.0 (Security Assertion Markup Language). To integrate with SAML SSO, you will need to place a user-specific link to LearningZen.com on your site (or intranet, etc.). You will also need to create a simple web service with which the LearningZen server can communicate. The web service will tell us if the user is currently logged in and will provide some basic information about that user. If the user already has an account inside LearningZen, their basic information will be updated. If it is a new user, a LearningZen account will automatically be created.

Configuration

To properly configure SSO, the Third Party (TP) must provide LearningZen with the following information:

- **Authentication Failure URL**: where we should send your users when authentication fails, presumably your log in page. e.g. https://third-party.com/login

- **SAML SSO X.509 Certificate PEM File**: the base URL of your web service. The API method name will be appended to the end of this URL when making an API call. e.g. https://third-party.com/api

- **SAML SSO URL**: the base URL where the AuthnRequest payload will be sent.

- **User Attribute Mappings:** These are attributes(<SamlAttribute>) about the user that will be updated upon successful authentication. This list uses LZ's internal names, but the "FriendlyName"s may be whatever is convenient for the IdP
    - **AccountID**(Required)
        - The primary identifier in the IdP
        - Less than or equal to 256 characters
    - **EmailAddress**(Required)
        - The address LZ should send any emails for the user to
    - **UserFirstName**(Required)
        - Less than or equal to 32 characters
    - **UserLastName**(Required)
        - Less than or equal to 32 characters
    - **UserGroups**(Optional)
        - Comma(,) delimited list of group names the user should belong to. The user will be removed from any groups not in this list. Group names cannot contain commas.
        - UserGroups & ManagerGroups Can be ignored via an internal configuration
    - **ManagerGroups**(Optional)

        - Comma(,) delimited list of group names the user should belong to. The user will be removed from any groups not in this list. Group names cannot contain commas. Can be ignored via an internal configuration
        - UserGroups & ManagerGroups Can be ignored via an internal configuration

    - **IsAuthor**(Optional)
        - Should the user have the “Author” role
        - “1” for true, or “0” for false
        - IsAuthor, IsManager & IsAdmin Can be ignored via an internal configuration

    - **IsManager**(Optional)
        - Should the user have the “Manager” role
        - “1” for true, or “0” for false
        - IsAuthor, IsManager & IsAdminCan be ignored via an internal configuration

    - **IsAdmin**(Optional)
        - Should the user have the “Admin” role
        - “1” for true, or “0” for false
        - IsAuthor, IsManager & IsAdmin Can be ignored via an internal configuration

    - **TimeZoneName**(Required)
        - The User’s timezone name. This will be used for Date & Time conversions. An incorrect value for the user could lead to incorrect dateTimes being displayed
        - Possible Values
        - https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/default-time-zones

Step-by-Step

The steps below walk you through the SSO mechanism:

Integration with SAML2.0 involves three entities; LearningZen (LZ, service provider), Identity Provider (Your System, IdP) and IdP User.

Step 1: The user requests a LZ URL or resource Ex.(/dashboard/dashboard).

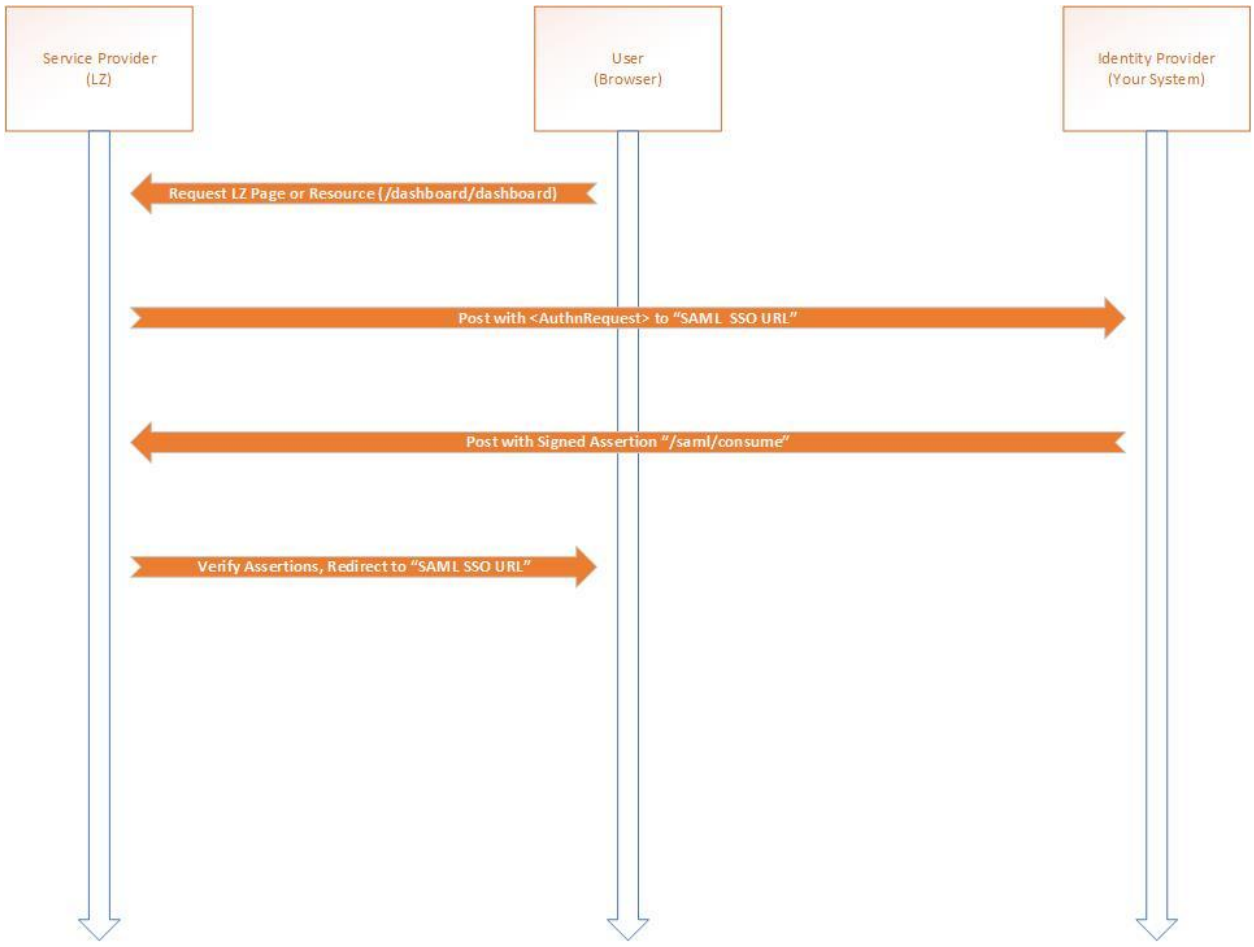
Step 2: To authenticate the user, LZ constructs a SAML AuthnRequest, encodes it then posts the same to theIdentity Provider (IdP) in order to authenticate. The IdP receives the request, decodes it and verifies the request. After validating the request, IdP will present the user with a login form (presumably if the user is not already logged into the IdP system).

Step 3: Once the user’s identity is verified on the IdP, the IdP will generate a SAML Assertion that includes the identity information about the user (User Attributes, such as userid, email, username etc), signes it with their certificate and encodes it. The IdP takes this SAML Response and redirects the user back to LZ (“/saml/consume”).

Step 4: LZ verifies the SAML Response, decodes it and extracts the identity information and validates the same.

Step 5: At this point, if the user has not been redirected to the Authentication Failure URL, they will be logged into LearningZen and will be redirected to the page they were originally trying to access. The user will remain logged into LearningZen until their LearningZen session is ended.

High Level



Technical Flow

LZ will post an HTML form to idP's "**SAML SSO URL**". This HTML FORM contains a SAML <AuthnRequest> Base64 encoded as the value of a hidden form control named SAMLRequest.

```
<form method="post" action="{SAML SSO URL}">
        <input type="hidden" name="SAMLRequest" value="{Base64 Encoded SAMLRequest Value}" />
        <input type="submit" value="Submit" />
</form>
```

The format of SAML AuthnRequest would be as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
 <samlp:AuthnRequest
AssertionConsumerServiceURL="{https://[YOURPORTAL].learningzen.com/saml/consume}"
Destination="{SAML SSO URL}"
ID="_9d646775171ab2d0351a5cde098484ba"
IssueInstant="2015-10-07T08:21:19Z"
ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Version="2.0"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">Learning Zen</saml:Issuer> </samlp:AuthnRequest>
```

Now the IdP will extract the AssertionConsumerServiceURL from the SAMLRequest, decode it from Base64 and validate the user's active session. Then, a SAML Response will be constructed, encoded to Base64, and sent to the AssertionConsumerServiceURL.
"RelayState" can be left Empty or removed from the response if desired.

Below are the sample code flow for the same:

```
<form method="post" name="sendSAMLResponseForm"
action="{https://[YOURPORTAL].learningzen.com/saml/consume}"  target="_blank">
<input type="hidden" name="SAMLResponse" value="{EncodedSamlResponseToken}" />
<input type="hidden" name="RelayState" value="" />
 </form>
```

The IdP will share the X.509 certificate with LZ. The IdP will digitally sign the SAML Response token and will append signed info with SAML token. Below is the format of sample Response token:

```
<?xml version="1.0" encoding="UTF-8"?> <samlp:Response
Destination="http://www.example.com/ProcessSamlResponsePage.aspx"
ID="_e14575012d8e77d134bdc825746d363f" IssueInstant="2015-10-07T08:22:53Z" Version="2.0"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"> <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">http://www.example.com/IDProvider/</saml:Issuer>
<samlp:Status> <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
```

```
</samlp:Status> <saml:Assertion ID="_432c584146858797a721ef712593be7a"
IssueInstant="2015-10-07T08:22:53Z" Version="2.0" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
<saml:Issuer>http://www.example.com/Idprovider/</saml:Issuer> <ds:Signature
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> <ds:SignedInfo> <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" /> <ds:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" /> <ds:Reference
URI="#_e14575012d8e77d134bdc825746d363f"> <ds:Transforms> <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" /> <ds:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" /> </ds:Transforms> <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<ds:DigestValue>nlpKIylgQHO797W0kZc2x9Rg8bg=</ds:DigestValue> </ds:Reference> </ds:SignedInfo>
<ds:SignatureValue>JHtuGC/cP3/bzL0wNIBfTWYcUGtkvBsOMD6ssnkxvExq1Z05FMOftoncsq++j25uNx0M
uWj2oV9M QuKnt8ax9ppm/xeNu+D2JquRa6ajWldqu6rW7/ycWk5lSxLJTVVmhqjNnkd5OpS9ehbfZr5JZ0TF
H2zoLqLfEjiG4PLs2xs=</ds:SignatureValue> <ds:KeyInfo> <ds:KeyValue> <ds:RSAKeyValue>
<ds:Modulus>sIRDnMAo7+FDxKUVNnW1TzKHL0dOsymhHV8a/USBOmVmEIHNrXPJUoDsIAQx6HSKcZP
mMmz2heYo+5EUv7/82Xu0GRXgwt
IB5wZzCJLtY5CmvD3w8lAQk4NmsMmT7fUlfCSolSUzqkEKiyYeFsCEg0FE 1. 2. 3. 4. 5. 6. 7.
w3/K95eKJdtCOyi4dYs=</ds:Modulus> <ds:Exponent>AQAB</ds:Exponent> </ds:RSAKeyValue>
</ds:KeyValue> </ds:KeyInfo> </ds:Signature> <saml:Subject>
<saml:NameID>Jane.Doe@domain.com</saml:NameID> <saml:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"> <saml:SubjectConfirmationData
NotOnOrAfter="2015-10-07T08:32:53Z"
Recipient="http://www.example.com/ProcessSamlResponsePage.aspx" /> </saml:SubjectConfirmation>
</saml:Subject> <saml:Conditions NotBefore="2015-10-07T08:12:53Z"
NotOnOrAfter="2015-10-07T08:32:53Z"> <saml:AudienceRestriction>
<saml:Audience>https://stage.serviceproviderURL.com</saml:Audience> </saml:AudienceRestriction>
</saml:Conditions> <saml:AuthnStatement AuthnInstant="2015-10-07T08:12:53Z"
SessionIndex="_1387325a93d9f54cb4e11e6f0f04141d" SessionNotOnOrAfter="2015-10-07T08:32:53Z">
<saml:AuthnContext>
<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</saml
:AuthnContextClassRef> </saml:AuthnContext> </saml:AuthnStatement> <saml:AttributeStatement>
<saml:Attribute Name="UserID" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
FriendlyName="UserID"> <saml:AttributeValue>12345</saml:AttributeValue> </saml:Attribute>
<saml:Attribute Name="UserName" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
FriendlyName="UserName"> <saml:AttributeValue>Jane Doe</saml:AttributeValue> </saml:Attribute>
<saml:Attribute Name="EmailAddress" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
FriendlyName="EmailAddress"> <saml:AttributeValue>Jane.Doe@domain.com</saml:AttributeValue>
</saml:Attribute> {Other User Attributes}.... </saml:AttributeStatement> </saml:Assertion>
</samlp:Response>
```


LZ will then validate the SAML response token claims sent from the IdP and will grant access to the destination URL/Resource if the response is successfully validated. If the response is not validated, then LZwill redirect the user to either the "**Authentication Failure URL**", or default to LZ's login page if not provided.