Yellowfin

connecting people and their data

Yellowfin SAML Bridge Web Application

2017

Introduction

The Yellowfin SAML Bridge is a Java web application that allows for interfacing between a SAML Identity Provider, and Yellowfin. This allows for a user to use the same credentials that they use for other applications at their organisation. The Yellowfin SAML Bridge in this case is a SAML Service Provider (SP). The SAML Bridge uses Yellowfin's web services to SSO the user into Yellowfin. There is also an option for auto provisioning users the first time that they connect to Yellowfin using the SAML bridge.

Installation

The Yellowfin SAML Bridge is a separate Java web application that can be run within the same Tomcat instance as Yellowfin. The bridge can be installed by unzipping the **YellowfinSAMLBridge.zip** file into the **Yellowfin/appserver/webapps/** directory.

The Yellowfin SAML Bridge uses the Yellowfin Webservice Java Library. The library (yfws.jar) that corresponds to the Yellowfin instance version should be included in the **/WEB-INF/lib** directory of the Yellowfin SAML Bridge.

SAML SP Configuration

The Yellowfin SAML Bridge uses the OneLogin Java API to interface with SAML Identity Providers (IDP). The configuration for the SAML SP is done within the **WEB-INF/classes/onelogin.saml.properties** file.

The following properties need to be set to configure the Service Provider (The Yellowfin SAML Bridge). There are inline comments with the properties file that give more information about each option.

onelogin.saml2.sp.entityid

This is the entityId of the SAML Bridge SP. This will be the metadata URL for SAML Bridge. The URL is of the form:

<scheme>://<host>:<port>/<context>/metadata.jsp

For example:

http://localhost:8080/samlbridge/metadata.jsp

The SP entityid may need to be registered with the SAML IDP to allow users access to this service.

onelogin.saml2.sp.assertion_consumer_service.url

This is the URL that handles a successful authentication.

The URL is of the form:

<scheme>://<host>:<port>/<context>/acs.jsp

For example:

http://localhost:8080/samlbridge/acs.jsp

onelogin.saml2.sp.single_logout_service.url

This is the URL that handles a logoff.

The URL is of the form:

<scheme>://<host>:<port>/<context>/sls.jsp

For example:

http://localhost:8080/samlbridge/sls.jsp

onelogin.saml2.sp.x509cert

This is the text representation of a security certificate. A self-signed certificate can be generated with:

```
openssl req -newkey rsa:2048 -new -x509 -days 3652 -nodes -out sp.crt -keyout sp.pem
```

The text representation of the sp.crt from the above command is required for this option.

onelogin.saml2.sp.privatekey

This is the text representation of the certificates private key. This is the text representation of the sp.pem file that was created by the self-signed certificate process above.

Any changes made to the **onelogin.saml.properties** file will require the Yellowfin SAML Bridge to be restarted for new settings to take affect.

SAML IDP Configuration

The Yellowfin SAML Bridge uses the OneLogin Java API to interface with SAML Identity Providers (IDP).

The configuration for the SAML IDP is also done within the

WEB-INF/classes/onelogin.saml.properties file.

Each SAML Identity Provider will require different options to be filled out in the properties file. A minimal configuration will require at least the following options to be filled, however depending on the IDP used, more options may be required:

onelogin.saml2.idp.entityid

onelogin.saml2.idp.single_sign_on_service.url

onelogin.saml2.idp.single_logout_service.url

The IDP Administrator should be able to provide the details for enabling a connection from the Yellowfin SAML Bridge to the desired IDP. The SP entityid may need to be registered with the SAML IDP to allow users access to this service.

Any changes made to the **onelogin.saml.properties** file will require the Yellowfin SAML Bridge to be restarted for new settings to take effect.

Yellowfin SAML Bridge Settings

Settings related to the operation of the SAML Bridge are located in the **WEB-INF/web.xml** file. These settings describe the location of the Yellowfin Instance and the web service credentials, and the attributes for finding and automatically provisioning Yellowfin Users.

YellowfinWebserviceURL

URL to the Yellowfin instance.

YellowfinWebserviceUser

Yellowfin User to use for connecting to Yellowfin's web services.

YellowfinWebservicePassword

Yellowfin User password for authenticating Yellowfin's web services.

YellowfinRole

The Yellowfin Role name, or code to assign to auto provisioned users.

AutoProvision

True or False. Provision users automatically when they log in via the Yellowfin SAML Bridge for the first time.

UsernameAttribute

The SAML attribute for the username to be used for syncing with Yellowfin. This is required even if auto-provisioning.

EmailAttribute

The SAML attribute for extracting the email address of the user. This is only required if auto-provisioning is on.

FirstNameAttribute

The SAML attribute for extracting the first name of the user. This is only required if auto-provisioning is on.

LastNameAttribute

The SAML attribute for extracting the last name of the user. This is only required if auto-provisioning is on.

FullNameAttribute

The SAML attribute for extracting the full name of the user. This will split the given name into first and last names, by splitting the name at location of the first space in the name. When this attribute is provided, the FirstNameAttribute and LastNameAttribute do not need to be set. This is only required if auto-provisioning is on.

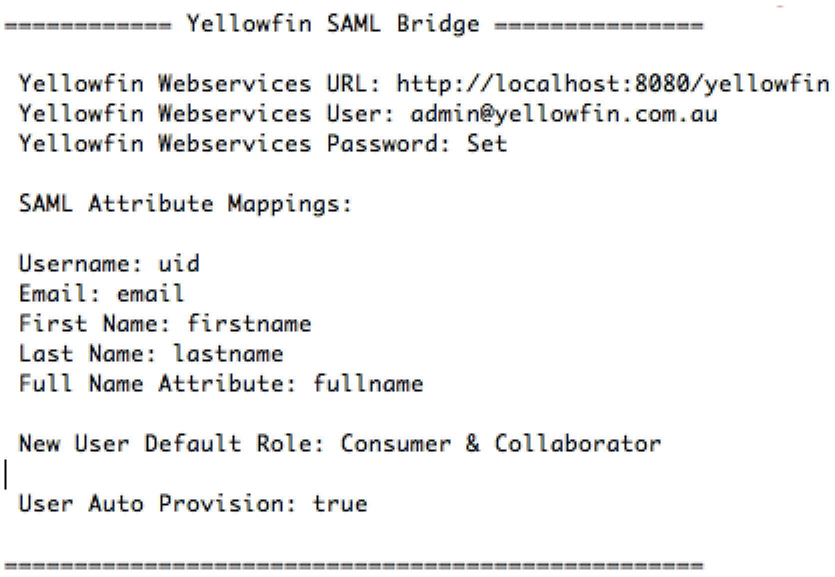
Any changes made to the web.xml will require the Yellowfin SAML Bridge to be restarted for new settings to take effect.

Yellowfin Instance Configuration

SIMPLE_AUTHENTICATION needs to be enabled to allow the Yellowfin SAML Bridge to login users. This option needs to be enabled to allow the Yellowfin SSO web service to login a user without a password.

Debugging Configuration Issues

The Yellowfin SAML Bridge will output the following information on startup:

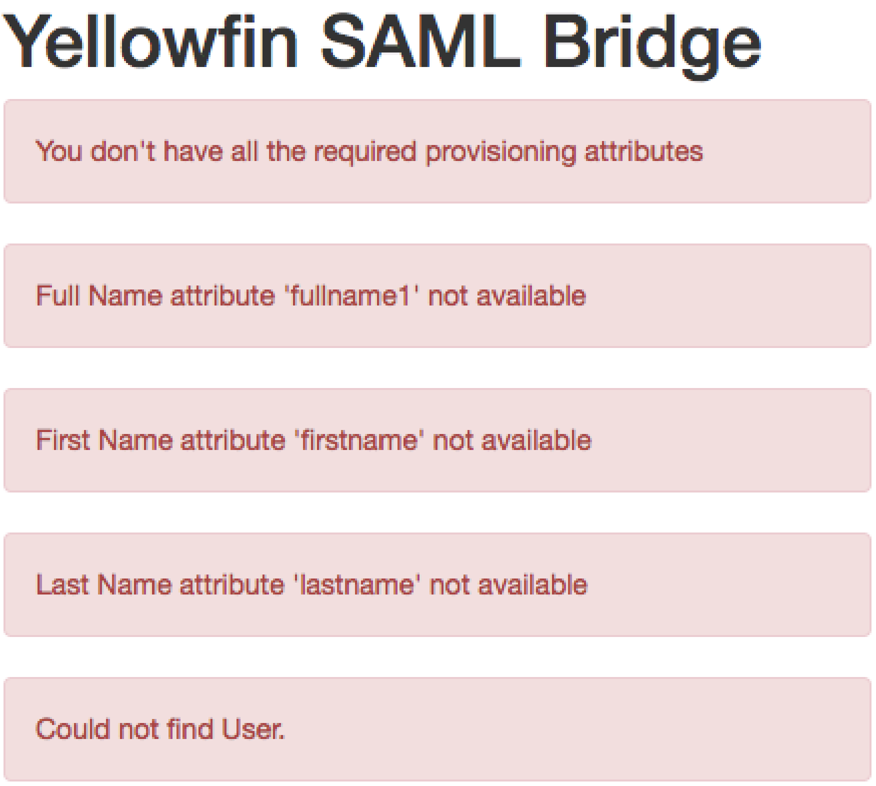


```
============ Yellowfin SAML Bridge ===============

 Yellowfin Webservices URL: http://localhost:8080/yellowfin
 Yellowfin Webservices User: admin@yellowfin.com.au
 Yellowfin Webservices Password: Set

 SAML Attribute Mappings:

 Username: uid
 Email: email
 First Name: firstname
 Last Name: lastname
 Full Name Attribute: fullname

 New User Default Role: Consumer & Collaborator

 User Auto Provision: true

==================================================
```

This will show which parameters have been detected at startup.

If there is a configuration error, or no matching attributes can be found when attempting to authenticate then errors will be shown from the acs.jsp page.

Here is an example:



Yellowfin SAML Bridge

You don't have all the required provisioning attributes

Full Name attribute 'fullname1' not available

First Name attribute 'firstname' not available

Last Name attribute 'lastname' not available

Could not find User.

Restyling

The JSP files included in the Yellowfin Bridge can be restyled to suit the organisation. Only HTML elements should be changed. Any modifications to the Java code could result SAML handshake not working correctly.

It may be required that the default Yellowfin Homepage be modified to fully integrate the Yellowfin SAML Bridge into the User Interface as desired. This could also be configured to perform a SAML logout when a Yellowfin session is destroyed.

Useful Links

The OneLogin Java API page, contains some information on the settings required for configuring the SP and IDP.

https://github.com/onelogin/java-saml