How to log a user in based on an existing session

or How to create an SSO session using an existing application session
or How to create a link to choose the authentication method on behalf of a a user

This page is incomplete in some areas. Please contact Ubisecure Support or Sales Engineering for more information if required.

Use Cases

Use case A: A user already has a session with an application and has been authenticated using some existing technique. How can that user be logged into the Ubisecure SSO system to enable single sign-on to other applications?

Use case B: Customer support need to log in to the system as the end-user, in order to help them solve a problem or to see the same view that the user sees. The customer support user already has a session with an application and has been authenticated using some existing technique. How can that user be logged into the Ubisecure SSO system as a different user to enable single sign-on to other applications?

Use case C: A user has been authenticated using non-browser based API, such as OAuth password grant. Because no browser is used, there is no session cookie created that would allow the user to perform browser-based single sign-on to another application.

In each of the above cases, it is possible to create a new browser-based SSO session using a technique called SAML unsolicited response. Ubisecure provides libraries and examples for Java and powershell to create a new SAML response based on existing information at hand.

The created SAML response is sent to a special endpoint that creates the session at the server and issues a new response to the target SAML service provider.

The name of the endpoint is SessionRelayService.

Attached is technical documentation in javadoc format for the Java library that implements the required functions. A sample implementation is included. The sample jsp page shows a simple drop down menu, from which the user can choose one of three users. After submitting the form, the user will be logged in as that user.

The application is a simple model only - a real implementation would require that the user is securely identified before impersonating another user.

Activation of the application will require the generation of the private and public key and metadata for activation. The ubisaml2.jar library provides this capability. It must be run with IDP role option.

Creating a SAML Identity Provider Configuration

Ubisecure HTTP Header Authentication Provider acts as SAML 2.0 identity provider.

Create a SAML identity provider configuration by running the following script containing base URL. Please note that the java command must be found in PATH to run this script.

Listing 3. Create SAML configuration

```
cd /d "C:\Program Files\Ubisecure\httpheaderap"
java -jar webapp\WEB-INF\lib\ubisaml2.jar Generate https://example.com/samlap -o webapp\WEB-INF\uap -y -disable
SingleLogoutService
```

As a result, identity provider configuration is created in the httpheaderap/webapp/WEB-INF/uap/identity.properties file.

Exporting SAML Identity Provider Metadata

Export the SAML identity provider metadata by running the following script.

Creating SAML metadata file

```
cd /d "C:\Program Files\Ubisecure\samlap"
java -jar webapp\WEB-INF\lib\ubisaml2.jar Metadata webapp\WEB-INF\uap -idp -f metadata.xml -y
```

As a result, the identity provider metadata is created in the samlap/metadata.xml file.

Import the SAML Authentication Provider metadata

1. Log into Ubisecure Server Management with System Administrator privileges.
2. Navigate to the Server Authentication Methods view.
3. Press Home -> Methods, Add new method
4. Title - human readable name "SAML AP Example"
5. Name - technical name for logging, "saml.ap.1"
6. Method Type - SAML
7. Method Class - Selected automatically
8. Directory - Use only if performing Directory User Mapping during login. Refer to Ubisecure Server Management Guide for more information.

Once created, select the Hidden checkbox and press update. This type of Authentication Method should not be selectable by the end-user, as the basic example supports only unsolicited response messages.

Choose the SAML tab and click Upload to upload the metadata of SAML AP generated in step XXX to the newly created SAML method. Select the XML file generated.

Choose the Main tab
Select Enable and press Update.

Open the settings of the authentication method you just installed and choose the SAML view. Click Upload SAML Metadata and either upload the metadata file samlap/metadata.xml created in section Exporting SAML identity provider metadata or copy and paste the metadata into the textarea shown.

Export the SAML service provider metadata

Click Download SAML Metadata to download the service provider metadata. This file must be saved to the application directory samlap/webapp/WEB-INF /uap/metadata using the filename sp-metadata.xml. In this case, the Ubisecure SSO server is operating in an SP role.

Enable the method for each site which contains applications that will use this method:

1. Select the site from the Site Navigator
2. Select the methods tab
3. Choose Add method...
4. Select the SAML AP Example method.

Enable the method for application that will allow users authenticated by the custom SAML Authentication Provider to log in.

1. Select the Application
2. Select the Methods tab
3. Select the check box for the SAML AP Example method
4. Press Update

IDP Initiated SSO using SAML2

An unsolicited SSO can be done by sending a valid SAML response message to the address:

```
https://www.example.com/uas/saml2/SessionRelayService?entityID=urn:uuid:3A97e9cf6b-5218-4cb8b0b9-
bab5d35e6c9b&RelayState=/insert/home/page/here&locale=sv
```

where:
-entityID has to be application objects entityID from Ubisecure SSO Management UI
-RelayState is relative address on target application server where browser is redirected(so called deep linking)
-locale is users used language

You can map this address to a nicer shorter URL using any other tools (by redirect).

SessionRelayService calls can also be chained:

```
https://sso.example.com/uas/saml2/SessionRelayService?entityID=https://sso.example.com/uas/saml2/names/ac/saml.
companyx.1&RelayState=/uas/saml2/SessionRelayService?entityID=urn:uuid:6c524df0-4625-32a8-87ef-705b3523e4b2%
26RelayState=/app/protected
```

WS-Federation Passive Requester Profile

The WS-Federation Passive Requester Profile is used for initiating a login request. A request is formed at the PassiveRequestorService endpoint:

```
https://www.example.com/uas/wsf/PassiveRequestorService?wa=wsignin1.0&wtrealm={entityID}
```

The available parameters are:

- wa is always ”wsignin1.0” (mandatory)
- wtrealm is the entityID of the target application (mandatory)
- wctx allows for the passing and return of RelayState (optional)
- wreply is not used by Ubisecure SSO (optional)
    - This OPTIONAL parameter specifies the URL to which responses will be sent. It must be a one of a list of pre-configured URL in the metadata.
- wauth
    - This OPTIONAL parameter indicates the REQUIRED authentication level. It is equiavelent to SAML2 AuthnContextClassRef.
        - e.g. wauth=urn:oasis:names:tc:SAML:2.0:ac:classes:Password
- whr (optional) specifies the desired authentication method (equiavelent to SAML2 AuthnContextDeclRef). It can be specified in short form (method name) or long form (URI)
    - whr=password.1
    - whr=https://sso.example.com:8443/uas/saml2/names/ac/password.1
- wfresh (optional)
    - This optional parameter indicates the freshness requirements. If specified, this indicates the desired maximum age of authentication specified in minutes. If specified as “0” it indicates a request for the IP/STS to re-prompt the user for authentication before issuing the token. This is equivalent to OAuth2 max_age or SAML2 NotOnOrBefore concepts.
- locale (optional)
    - locale=fi
- template (optional)
    - Set the user interface template
    - template=christmas

Example 1 - requesting Finnish language Christmas template login with forced reauthentication and password.1 method only

```
https://www.example.com/uas/wsf/PassiveRequestorService?wa=wsignin1.0&wtrealm={entityID}
&locale=sv&template=christmas&wfresh=0&whr=password.1
```

Example 2 - requesting Finnish language Christmas template login with any valid session using any method that has the class of urn:oasis:names:tc:SAML:2.0:ac:classes:Password. If there is more than one method, a selection menu will be shown

```
https://www.example.com/uas/wsf/PassiveRequestorService?wa=wsignin1.0&wtrealm={entityID}
&locale=sv&template=christmas&wauth=urn:oasis:names:tc:SAML:2.0:ac:classes:Password
```

If a user has a session and is permitted to use the application, the user will be redirect to the application with a valid assertion.

Because the WS-Federation request is not signed and is thus easily spoofed by any party, the integrated application should check and compare each value of the response to ensure it met the requested parameters.

OAuth2 Applications

For OAuth2 applications, use the Authorization Request URL to initiate the process and acr_values to select the desired authentication method.

```
https://sso.example.com/uas/oauth2/authorization?response_type=code&scope=openid&client_id=2001221477
&redirect_uri=https://client.example.com/response&state=40e1bfc0-4587-4859-be08-
a58e3fffa37a&max_age=0&prompt=login&display=popup&ui_locales=en&acr_values=2&login_hint=user@example.com
```

Related articles

- Use an unsolicited SSO or an IDP initiated SSO
- SAML Compatibility Flags
- How to log a user in based on an existing session
- 1. Accessing logs and adjusting logging levels of SSO and CustomerID

- Change hostname of Ubisecure SSO