Configure Cora SeQuence for SAML 2.0 SSO with OAuth 2.0 for Service-to-Service

Last Modified on 12/18/2018 6:56 am EST

v8.4 and later

Overview

Learn how to configure Cora SeQuence Administration site and Flowtime Portal to allow service-to-service requests for Web Service Listeners and OData services.

When you configure Cora SeQuence Administration or Flowtime Portal using OAuth, all user authentication is executed using SSO (see Configuring Sequence to use SAML protocol and WS-Federation). You cannot use Windows authentication or Forms authentication. If you apply this configuration to an existing environment you might need to update existing code and any other applications that access Web Service Listeners and OData.

Introduction

Cora SeQuence can authenticate a request to Web Service Listener or an OData endpoint using OAuth bearer token that is sent by the client in the Authorization Header of an HTTP request. . The token should be acquired by the client prior to a calling to Cora SeQuence and it is the clients responsibility keeping the token secure.

All requests to *[Sequence URL]/SequenceServices* are inspected for a bearer token. If a bearer token is found, the token is validated. If the token is valid, the identity supplied by the token will be processed by Cora SeQuence authentication, and if the identity is an existing Cora SeQuence user, execution will continue with that user.

If the identity supplied by the token does not match any Cora SeQuence user, a 401 HTTP Response is sent to the caller.

Requests sent to URLs other the *SequenceServices* will not be inspected for a bearer token and will pass through a Single Sign-On flow (read the Single Sign-On article for more information). You can apply the same settings as on *SequenceServices* to other locations under the web application using the element in the `web.config` file.

Supported Scenarios

Prerequisites

- Configure SSO using SAML or WS-Federation to allow users's authentication.
- Have a configured Identity Provider that can issue bearer tokens to the client and have Cora SeQuence is registered as a Service Provider (Relying Party).

How to Configure

IIS Configuration

In order to allow authentication using a bearer token in Cora SeQuence the following should be configured in IIS

For Administration Web Application and Flowtime Portal:

- Make sure only anonymous authentication is set.
- Configure `Default.aspx` file as the default document for the root of the web application

web.config

The following sections of the web.config file should be modified

- configSections
- sequence.engine/authentication (root location)
- sequence.engine/identity (root location)
- sequence.engine (SequenceServices location)
- system.webServer/modules
- system.serviceModel (Administration Only)
- system.serviceModel (Flowtime Portal Only)
- system.identityModel

configSections

Under the configSections you need to add the *identity* as a sectionGroup and as sections under it the *oauth* and in this document will also add the *saml* section to demonstrate fully functional identity and authentication configuration.

```
...

 ...
 ...









 ...
 ...



 ...

...
```

sequence.engine/authentication (root location)

```
...
 ...
 ...
 ...

...
```

sequence.engine/identity (root location)

```
 ...
 ...
 ...
 ...
```

OAuth Configuration Options

Name	Description	Default Value	Required

Name	Description	Default Value	Required
enabled	When enabled any requests that passes through the *BearerTokenAuthenticationModule HttpModule* will be inspected, and when a bearer token will be found the module will try to read it and use it to authenticate the user against Cora SeQuence.	True	No
validIssuer	The name of the token issuer. The name will be validated against the token and signing certificate.	N/A	Yes
authority	The OAuth token endpoint.		
discoveryKind	The *discoveryKind* attribute sets how OAuth discover its identity provider's signing tokens.	OpenIdDiscoveryDocument	Yes
discoveryUri	When the discoveryKind is set to *OpenIdDiscoveryDocument* or *JsonWebKeySet* or *Federation*, this settings is used to locate the document containing the needed information to retrieve the identity provider's signing keys	Empty String	Yes
validAudience	The audience for which a token is issued to. This value is usually a URI or any unique identifier in a string format.	Empty String	Yes

Name	Description	Default Value	Required
validateAudience	Indicates whether the audience in the token should be validated against the audience set in this configuration	True	No
validateIssuer	Indicates whether the issuer of the token should be validated against the issuer set in this configuration.	True	No
validateIssuerSigningKey	Indicates whether to validate that the token is signed by the issuer set in this configuration.	False	No
identityProviderCertificates	A collection of configuration elements setting how to locate a certificate in the machine Certificate Store. This element is required only when the discoveryKind is set to 'Store'	Null	No

discoveryKind options

A token is usually signed by the Identity Provider using a Public Certificate. Cora SeQuence should be able to obtain the Public Certificate in order to validate the Identity Provider's signature on the token. There are four ways that Cora SeQuence can discover the public certificate:

discoveryKind Attribute	Description
OpenIdDiscoveryDocument	Indicates that the discovery will be done using an OpenIdDiscoveryDocument . When this options is set, the discoveryUri attribute must be set to a Uri that contains this type of document.
JsonWebKeySet	This option is expecting a Uri that returns a JsonWebKeySet result. This settings is a sub setting of the OpenIdDiscoveryDocument.
Federation	This option expect is expecting a Uri that returns a WS-Federation metadata document. When this options is set, the discoveryUri attribute must be set to a Uri that contains this type of document .

discoveryKind Attribute	Description
Store	This option is used when the certificates are stored on the machines local certificate store. When this option is set the *IdentityProviderCertificates* element is required.

identityProviderCertificate options

Under this element you configure where and how to retrieve certificates from the local certificate store. This is relevant only when setting *Store* under the *discoveryKind* attribte.

Attribute	Description
name	A unique name for each certificate
certificateFindType	This value is based on the *System.Security.Cryptography.X509Certificates.X509FindType* .
certificateFindValue	This value should contain the actual value that will be used to find the certificate based on the *certificateFindType* attribute.
certificateLocation	This value is based on the *System.Security.Cryptography.X509Certificates.StoreLocation* enum.
valid	Indicates if the certificate can be retrieved even if it is not valid (entire chain can not be validated).

sequence.engine (SequenceServices location)

Under the location of SequenceService you should add the entire element and its child elements.

```
...
...
```

system.webServer/modules

- Make sure the modules below are present and ordered as shown. If you already have other modules registered under this element, make sure the new modules are ordered under the existing Cora SeQuence related modules.
- Verify that the existing SequenceAuthenticationModule element has the *preCondition="managedHandler"* attribute set.

```
...
...
...

...
...
...
```

system.serviceModel (Administration Only)

Make sure that the endpoint binding for DataStreamDesignService is configured to use HTTPS. See the following example.

```
...
...
```

system.serviceModel (Flowtime Portal Only)

Make sure that the endpoint binding is configured to use HTTPS. See the following example.

```
...
...
```

system.identityModel

```
...



...
```