

IBM Security Secret Server

SOAP Web Services Guide

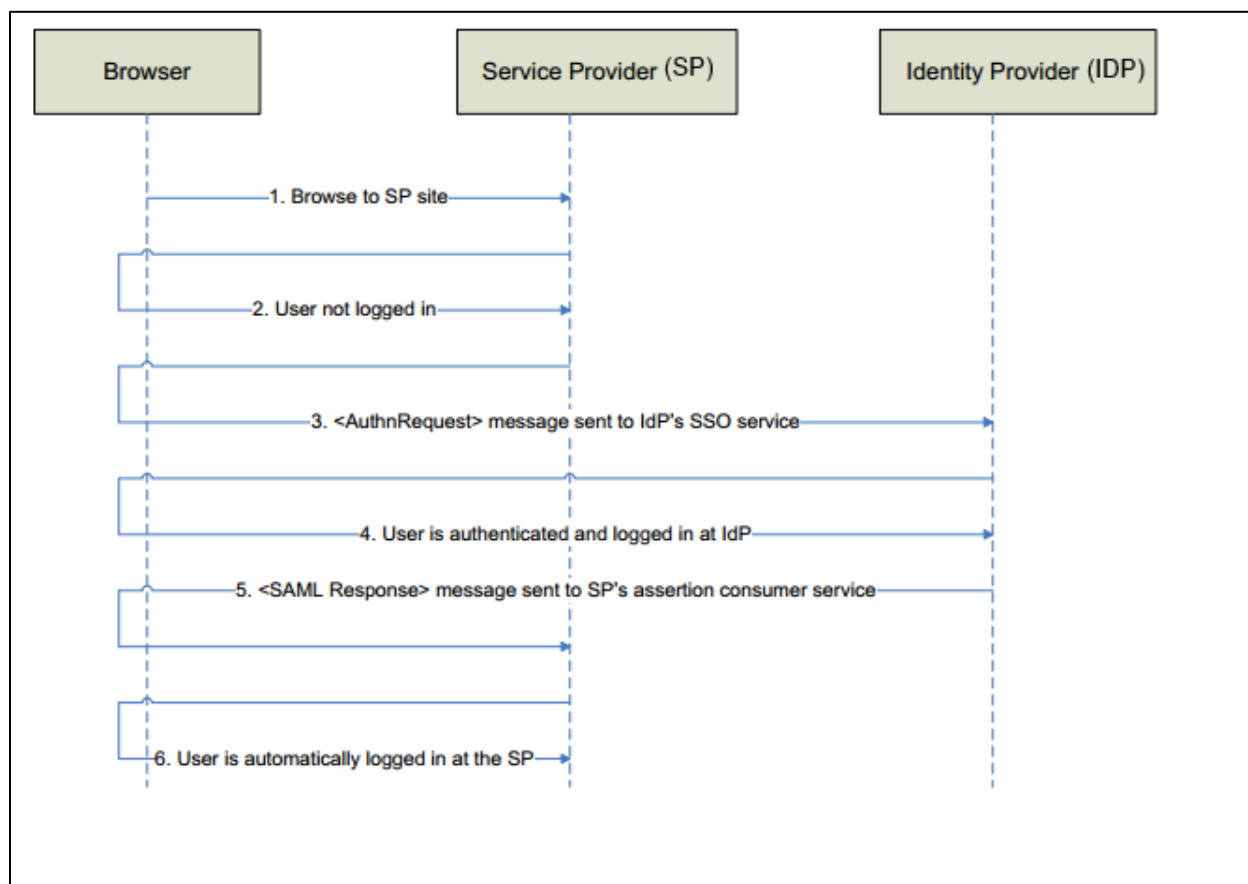
Table of Contents

- SAML Authentication with Secret Server1**
- Configure SAML.....1**
 - Secret Server2
 - SAML File Configuration2
- How to Specify a Certificate for SAML5**
 - How to specify a SAML Certificate on the File System6
 - How to specify a SAML Certificate in the Windows Certificate Store6
 - Example SAML config file for a SimpleSAMLPHP installation:6

SAML Authentication with Secret Server

Secret Server allows the use of SAML Identity Provider (IdP) authentication instead of the normal authentication process for single sign-on (SSO). To do this, Secret Server acts as a SAML Service Provider (SP) that can communicate with any configured SAML IdP. The documentation below assumes you have a running IdP with a signed certificate.

In the diagram below, Secret Server acts as the Service Provider. Any configured SAML Identity Provider can be used for this process and there are several well tested providers, including Shibboleth, SimpleSAMLPHP and Microsoft ADFS.



Configure SAML

SAML Configuration in Secret Server consist of two parts: Enabling SAML in the Secret Server UI and modifying the saml.config file for Identity Provider specific settings.

SECRET SERVER

1. Go to **ADMIN | Configuration** and click the **Login** tab. Click the Edit button and check the box for **Enable SAML Integration**.

The screenshot shows the 'Login' configuration tab in the Secret Server administration interface. The 'Enable SAML Integration' checkbox is checked and highlighted with a red box. The 'SAML Username Attribute' field is empty. Other visible settings include 'Allow Remember Me', 'Allow Two Factor Remember Me', 'Allow AutoComplete', 'Maximum Login Failures' (5), 'Enable Login Failure CAPTCHA', 'Default Login Domain' (Last Selected), 'Maximum concurrent logins per user' (1), 'Visual Encrypted Keyboard Enabled', 'Visual Encrypted Keyboard Required', 'Require Two Factor for these Login Types' (Website and Web Service Login), 'Enable RADIUS Integration', 'Enable Duo Integration' (checked), 'API Hostname' (api-e5972877.duosecurity.c), 'Integration Key' (D13H89T3IKHSY7JBAI8M), and 'Secret Key' (masked with dots).

2. **Optional:** Specify the SAML Username Attribute. Most IdP's will pass the username in the 'Subject' of the Assertion / Response. This can typically be left blank.

SAML FILE CONFIGURATION

There are three parts to the backend SAML configuration. After following these steps, recycle the Secret Server application pool so the settings take effect.

1. Create the saml.config file
2. Modify the saml.config with your IdP settings
3. Modify the IdP's metadata to contain information on Secret Server as a Service Provider

Create the saml.config File

1. In the Secret Server directory (typically c:\inetpub\wwwroot\secretserver) copy the saml.config.template to a new file and name the new file saml.config.
2. Run notepad as an admin and edit the saml.config file.

Modify the saml.config With IdP Settings

Service Provider

Modify the Secret Server SAML configuration file for your ServiceProvider settings.

1. Choose a ServiceProvider EntityId. Typically this is the Secret Server URL and goes in the **Name** attribute of the ServiceProvider node.
2. The **AssertionConsumerServiceUrl** is a relative URL in Secret Server that should not need modification. Verify that it is: **AssertionConsumerServiceUrl="~/SAML/AssertionConsumerService.aspx"**
3. **[Optional]** Specify a Certificate to be used for signing a request during a **SP-Initiated login**. See the [How to Specify a Certificate for SAML](#) section for more information on using certificates.

Identity Provider

Fill out the PartnerIdentityProvider section in the saml.config file. Secret Server only supports one identity provider at a time.

1. Specify the EntityId of the Identity Provider in the Name attribute.
2. Specify the SingleSignOnServiceUrl (the URL on the IdP where users go to sign in).
3. Specify the SingleLogoutServiceUrl (the URL on the IdP where users go to sign out).
4. Specify additional options, such as encryption and signing:
 - a. **ClockSkew**
 - i. The optional ClockSkew attribute specifies the time difference allowed between local and partner computer clocks when checking time intervals. The default is no clock skew.
 - b. **DataEncryptionMethod**
 - i. The optional DataEncryptionMethod attribute specifies the XML encryption data encryption method. The default is: `http://www.w3.org/2001/04/xmlenc#aes128-cbc`.
 - c. **DigestMethod**
 - i. The optional DigestMethod attribute specifies the XML signature digest method. The default is: `http://www.w3.org/2000/09/xmldsig#sha256`.
 - d. **DisableInboundLogout**
 - i. The optional DisableInboundLogout attribute specifies whether logout requests sent by the partner provider are not supported. The default is false.
 - e. **DisableOutboundLogout**

- i. The optional `DisableOutboundLogout` attribute specifies whether logout requests sent to the partner provider are not supported. The default is false.
- f. ForceAuthn**
 - i. The optional `ForceAuthn` attribute is included in the authentication request, which requires the Identity Provider to re-authenticate, regardless of whether there is an existing session. The default is false.
- g. KeyEncryptionMethod**
 - i. The optional `KeyEncryptionMethod` attribute specifies the XML encryption key encryption method. The default is: `http://www.w3.org/2001/04/xmlenc#rsa-1_5`.
- h. LogoutRequestLifeTime**
 - i. The optional `LogoutRequestLifeTime` attribute specifies the `NotOnOrAfter` time interval for the logout request. The format is `hh:mm:ss`. The default is 3 minutes.
- i. SignatureMethod**
 - i. The optional `SignatureMethod` attribute specifies the XML signature method. The default is: `http://www.w3.org/2000/09/xmldsig#rsa-sha256`.
- j. SignLogoutRequest**
 - i. The optional `SignLogoutRequest` attribute specifies whether logout requests sent to the partner provider should be signed. The default is false.
- k. SignLogoutResponse**
 - i. The optional `SignLogoutResponse` attribute specifies whether logout responses sent to the partner provider should be signed. The default is false.
- l. SingleLogoutServiceBinding**
 - i. The optional `SingleLogoutServiceBinding` attribute specifies the transport binding to use when sending logout messages to the partner provider's SLO service. The default is to use the HTTP-Redirect binding.
- m. UseEmbeddedCertificate**
 - i. The optional `UseEmbeddedCertificate` attribute specifies whether the certificate embedded in the XML signature should be used when verifying the signature. If false, a configured certificate retrieved from the certificate manager is used. The default is false.
- n. WantLogoutRequestSigned**
 - i. The optional `WantLogoutRequestSigned` attribute specifies whether the logout request from the partner provider should be signed. The default is false.
- o. WantLogoutResponseSigned**
 - i. The optional `WantLogoutResponseSigned` attribute specifies whether the logout response from the partner provider should be signed. The default is false.

Modify IdP Metadata for Secret Server

Following the instructions provided by your Identity Provider (i.e. ADFS, SimpleSAML, Okta, etc...), add the appropriate entries for Secret Server as a Service Provider.

- Secret Server's assertion consumer service is located at: `https://<PATH TO YOUR SECRET SERVER>/SAML/AssertionConsumerService.aspx`.

- Secret Server's SingleLogoutService is located at: <https://<PATH TO YOUR SECRET SERVER>/SAML/sloservice.aspx>.
- Secret Server's EntityId (or URN or other similar reference) is the EntityId chosen in the section [Create saml.config file](#).

Example SAML Configuration File for SimpleSAMLPHP

```
<?xml version="1.0"?>
<SAMLConfigurations ReloadOnConfigurationChange="false"
xmlns="urn:componentspace:SAML:2.0:configuration">
<SAMLConfiguration>
<ServiceProvider
Name="urn:componentspace:SecretServerServiceProvider"
AssertionConsumerServiceUrl="~/SAML/AssertionConsumerService.aspx"
LocalCertificateFile="sp.pfx"
LocalCertificatePassword="password"/>
<PartnerIdentityProviders>
<PartnerIdentityProvider Name=https://localhost/simplesaml/saml2/idp/metadata.php
SignAuthnRequest="false"
WantSAMLResponseSigned="true"
WantAssertionSigned="false"
WantAssertionEncrypted="false"
SingleSignOnServiceUrl="https://localhost/simplesaml/saml2/idp/SSOService.php?spentityid=urn:componen
tspace:SecretServerServiceProvider"
PartnerCertificateFile="simplesaml.crt"/>
</PartnerIdentityProviders>
</SAMLConfiguration>
</SAMLConfigurations>
```

How to Specify a Certificate for SAML

X.509 certificates are used for XML signatures and XML encryption. A certificate for SAML can be specified in several ways within the saml.config file. A certificate may be stored in a file or the Windows certificate store.

The Certificate parameters should be prefixed with either **Local** or **Partner** depending if the certificate is for Secret Server to sign requests or the IdP signing assertions.

Support for authentication request signing using SHA-2

If Secret Server is running on .NET 4.6.1 or earlier, the local certificate must use the "Microsoft Enhanced RSA and AES Cryptographic Provider" Cryptographic Service Provider (CSP).

If Secret Server is running on .NET 4.6.2 or later, most legacy CSPs will properly function with SHA-2 signing.

HOW TO SPECIFY A SAML CERTIFICATE ON THE FILE SYSTEM

1. Specify a LocalCertificateFile or PartnerCertificateFile depending on if it is for the IdP signing the response or Secret Server signing the request. This can be an absolute path or a path relative to the application folder.
2. **[optional]** Specify a LocalCertificatePassword. This is the password associated with the certificate file. Certificate files (*.pfx) that include the private key should be protected by a password. For a production certificate, the password should be stored encrypted in web.config. Refer to the CertificatePasswordKey attribute directly below for more details.
3. **[optional]** Specify a LocalCertificatePasswordKey. This specifies the web.config's appSettings key for the certificate file password. For example, if the CertificatePasswordKey attribute value is localCertificatePassword, then under the web.config's appSettings section, an entry with the key name localCertificatePassword is expected and the entry value is used as the password. By encrypting the appSettings section using the aspnet_regiis utility, the certificate file password is secured.

HOW TO SPECIFY A SAML CERTIFICATE IN THE WINDOWS CERTIFICATE STORE

One of the following methods must be used to reference the certificate.

1. **[optional]** Specify a LocalCertificateSerialNumber or PartnerCertificateSerialNumber attribute. Specifies the X.509 certificate by serial number within the certificate store.
2. **[optional]** Specify a LocalCertificateThumbprint or PartnerCertificateThumbprint attribute. Specifies the X.509 certificate by thumbprint within the certificate store.
3. **[optional]** Specify a LocalCertificateSubject or PartnerCertificateSubject attribute. Specifies the X.509 certificate by subject within the certificate store.
4. **[optional]** Specify a LocalCertificateStoreLocation or PartnerCertificateCertificateStoreLocation attribute. Specifies the X.509 certificate store (LocalMachine or CurrentUser). The default is local machine.

EXAMPLE SAML CONFIG FILE FOR A SIMPLESAMLPHP INSTALLATION: