Troubleshooting Federation with Fiddler

Federation is a three person dance; there is the Identity Provider (IdP), the Relying Party (RP or SP), and the browser used to perform the federation. The hardest part of troubleshooting is often determining whether the error is on the Identity Provider, the Relying Party, or the browser. Fiddler, a free web debugging tool from Telerik, can be an invaluable tool for debugging federation issues.

Optimal IdM Support asked me to Capture a Fiddler Trace, what do I do?

If Optimal IdM Support has asked you to submit a Fiddler trace, this document will show you how to do that. Do the following:

1) Download and configure Fiddler (see *Downloading and Configuring Fiddler*).
2) Replicate the issue while running Fiddler with capture turned on. It helps to close all other web applications while doing this. Make sure to turn off Fiddler capturing once the error has been replicated.
3) Export the Fiddler Trace (see *Exporting a Fiddler Trace*) and attach it to your help desk ticket.
4) See the sections on debugging federation issues specific to your federation protocol for self-diagnostics.
    a. General troubleshooting tips - *Diagnosing General Federation Issues*
    b. WS-Federation – *Diagnosing Common WS-Federation Federation Issues*
    c. SAML 2.0 – *Diagnosing Common SAML2 Federation Issues*
    d. OAuth2\OpenID Connect – *Diagnosing Common OAuth2\OpenID Connect Federation Issues*

I am not using Optimal IdM but I still want to fix my federation problem, what do I do?

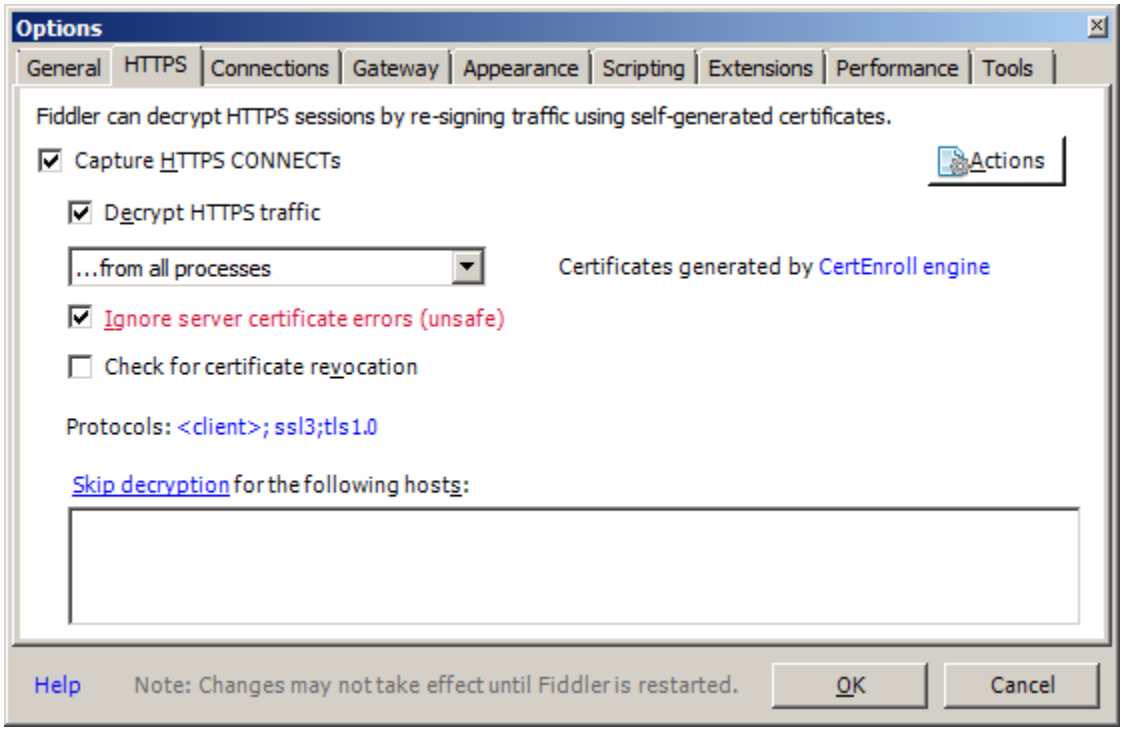
We are sorry you are using inferior technology. However, this document can still help you. Do the following:

1) Download and configure Fiddler (see *Downloading and Configuring Fiddler*).
2) Replicate the issue while running Fiddler with capture turned on. It helps to close all other web applications while doing this. Make sure to turn off Fiddler capturing once the error has been replicated.
3) See the sections on debugging federation issues specific to your federation protocol.
    a. General troubleshooting tips - *Diagnosing General Federation Issues*
    b. WS-Federation – *Diagnosing Common WS-Federation Federation Issues*
    c. SAML 2.0 – *Diagnosing Common SAML2 Federation Issues*
    d. OAuth2\OpenID Connect – *Diagnosing Common OAuth2\OpenID Connect Federation Issues*

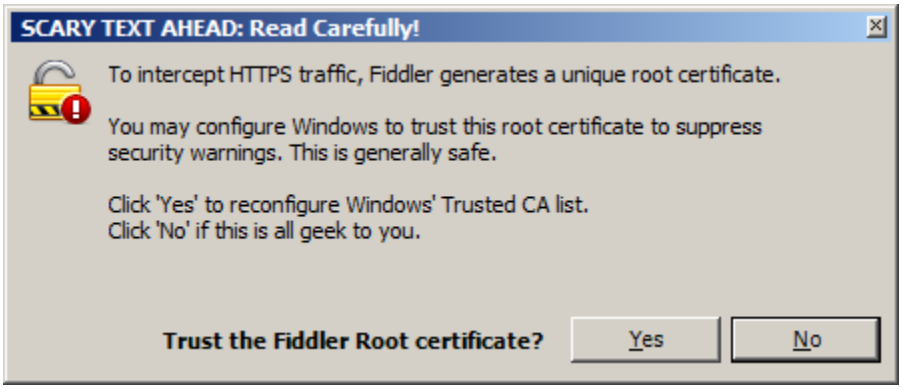
Downloading and Configuring Fiddler

Download the latest version of Fiddler from https://www.telerik.com/download/fiddler. After downloading, you need to configuring Fiddler for decrypting HTTP/SSL traffic.

After starting Fiddler, go to Tools -> Options and then select the HTTPS tab. Select “Capture HTTPS CONNECTs” and “Decrypt HTTPS traffic” options.



When you select this option, Fiddler will ask for permission to install dummy root certificates in you certificate store.



These certificates can be safely removed after using Fiddler to debug the federation issues.

If you have deployed Fiddler and used it to successfully decrypt HTTPS traffic but it suddenly stops decrypting HTTPS traffic, the dummy root certificates have likely expired. To resolve this

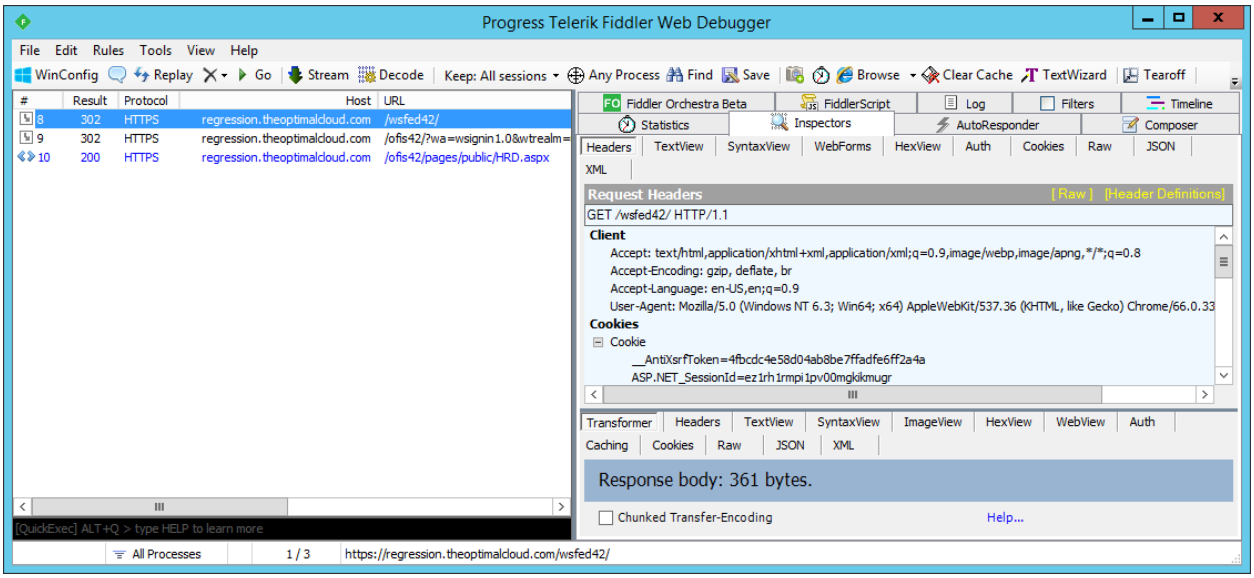
issue; turn off HTTPS decryption, remove all the dummy root certificates from your personal certificate store, and then enabled HTTPS decryption.

Capturing a Fiddler Trace

To capture a Fiddler trace, simply launch Fiddler and then go through the steps to reproduce the error.

Important! Once you replicate the issue with Fiddler running, make sure to turn off traffic capturing. That can be done via the menu (File -> Capture Traffic) or by clicking on the bottom left hand corner of the Fiddler application.

The Fiddler console will show all of the browser actions involved in the Federation and allow you to inspect each request and response.



If the only values you see in the Host column are “Tunnel To”, then you do not have HTTPS decryption properly. See *Downloading and Configuring Fiddler* for more information.

Exporting a Fiddler Trace

After capturing the Fiddler trace, you can save it by going to File -> Save -> All Sessions… Enter a file name and the Fiddler trace will be saved as a *.saz file. This is the file that should be sent to Optimal IdM support.

Diagnosing General Federation Issues

The URI for a relying party or identity provider may be in the form of a URL (such as http://my.test.com) or a URN (urn:my.test.com). URIs (both URNs and URLs) are case sensitive when used for Federation. For URLs in the form of URIs, every "/" is part of the name as is the protocol. When used as a URI the URLs http://my.test.com, http://my.test.com/, https://my.test.com, and https://my.test.com/ would all be considered different URIs. This often causes federation errors.

After capturing the Fiddler trace look for HTTP Response codes with value 404. The response code is the second column from the left by default and response code will typically be highlighted in red. If you see a 404 error there is like one of two reasons; 1) the URL is wrong and does not point to a valid location or 2) the URL length exceeds that which the server can support.

If you see a 404 error in the browser that does not show up in the Fiddler trace then that indicates the URL length exceeds the URL length limit of your browser. Browser URL length limits are vendor dependent.

Diagnosing Common WS-Federation Federation Issues

The basic flow of WS-Federation is:

1) The user requests an access to a relying party
2) The user is redirected to the Identity Provider (IdP) with a WS-Federation authentication request
3) The user then authenticates at the IdP
4) A WS-Federation authentication response is then posted to the relying party

Error on authentication request to the Identity Provider

If in the authentication request you get an error on the Identity Provider indicating that the relying party URI is not recognized, run a Fiddler trace reproducing the issue. Then look for a GET request to the IdP with the following URL parameters shown below. You can see the URL parameters by selecting the line in the request list and then going to the Inspectors -> Web Forms tab. The URL parameters for the WS-Fed authentication request are:

- wa = wsignin1.0
- wtrealm = <relying party URI>
- wctx = <federation context>
- wct = <request time in UTC>

Make sure the value in the wtrealm URL parameter matches the value configured at the identity provider for the relying party. Also look at the raw URL which can be seen on the Inspectors -> Raw tab. Make sure the wtrealm URL parameter is properly URL encoded in the raw authentication request URL.

Error on authentication response to the Relying Party

The WS-Federation reponse is an HTTP POST request with the follow form data. You can see the form data by selecting the line in the request list and then going to the Inspectors -> Web Forms tab. The form data for the WS-Fed authentication response are:

- wa = wsignin1.0
- wresult = <WS-Fed response XML>
- wctx = <federation context> (should match the authentication request)

From the WS-Fed response XML validate the following:

- Make sure the IdP URI matches the value configured on the relying party. This value can be found in the WS-Fed response XML Issuer element.
- Make sure the signing certificate matches the signing certificate configured on the relying party. The signing certificate can found in the WS-Fed response XML in the Certificate element under the Signature element.
- Make sure the assertion audience matches the relying party URI. The assertion audience can be found in the Audience element of the assertion.
- WS-Federation supports either SAML 1.1 or SAML2 assertions, make sure the type of the assertion matches what the relying party is expecting. For SAML 1.1 assertions the major and minor version attributes on the Assertion element will be set to 1. For SAML2 assertions the version attribute on the Assertion element will be set to 2.

Diagnosing Common SAML 2.0 Federation Issues

The basic flow of SAML 2.0 is:

1) The user requests an access to a relying party
2) The user is redirected to the Identity Provider (IdP) with a SAML 2.0 authentication request
3) The user then authenticates at the IdP
4) A SAML 2.0 authentication response is then posted to the relying party

While the basic flow is the same as WS-Federation, SAML 2.0 is much more complicated because the authentication request is an XML document rather and URL parameters. Also SAML 2.0 supports different methods of transporting the authentication request and response. These methods are called “Bindings”. The three most common bindings are POST, Redirect, and Artifact. The most common combination is for the authentication request to be passed using the Redirect Binding and the response is returned using the POST Binding.

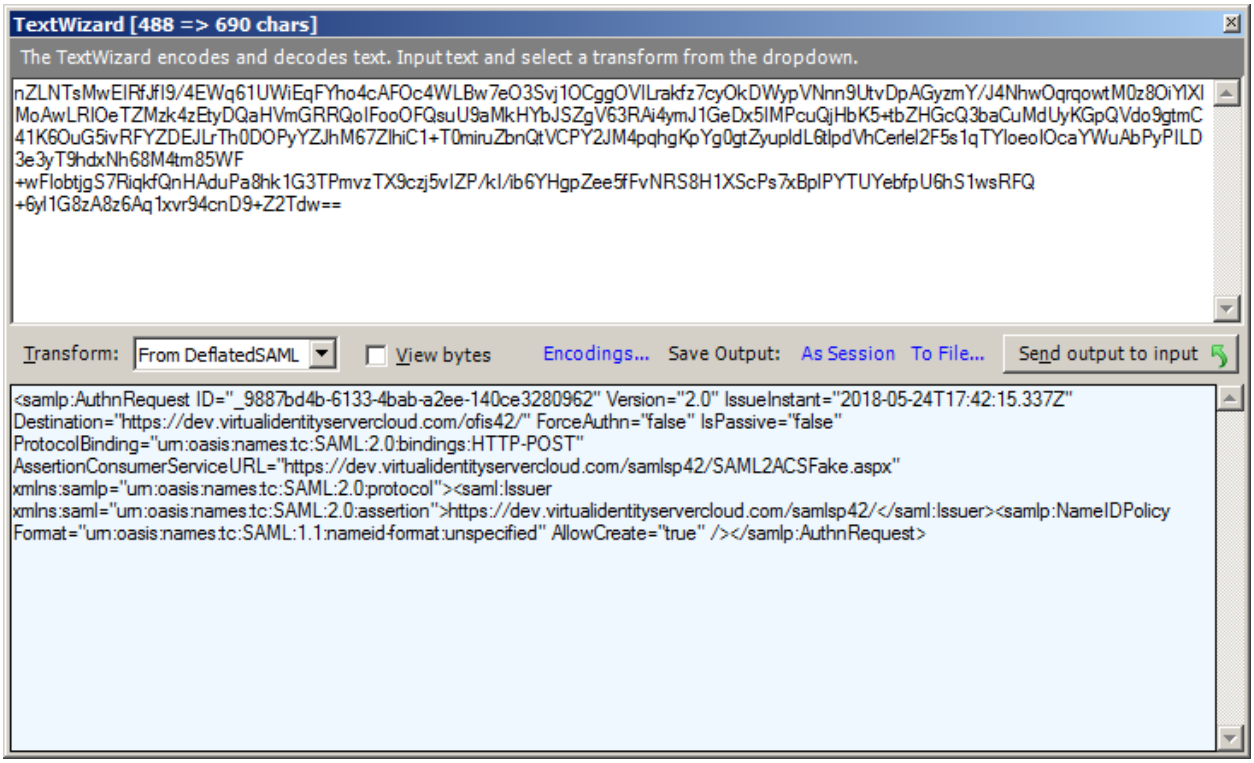
Error on authentication request to the Identity Provider

If you get an error on the authentication request to the IdP, capture a Fiddler trace reproducing the issue. Then look for a GET request to the IdP with the following URL parameters shown below:

- SAMLRequest – encoded SAML 2.0 Authentication Request XML

- SigAlg – XML Digital Signature Algorithm (optional)
- Signature – XML Digital Signature (optional)

Unlike WS-Fedation, the SAML 2.0 authentication request is an XML document that is compressed and encoded. Fortunately Fiddler can easily decode this for you and show you the XML document. Simply right click on the SAMLRequest value and select “Send to TextWizard …” That will bring up the Fiddler TextWizard window. If you don’t see XML make sure the Transform: drop down in the middle is set to “From DeflatedSAML”.



Check the following:

- Make sure the request Issuer value matches the relying party URI configured in the IdP.
- Make sure the Destination attribute matches the IdP SSO endpoint.

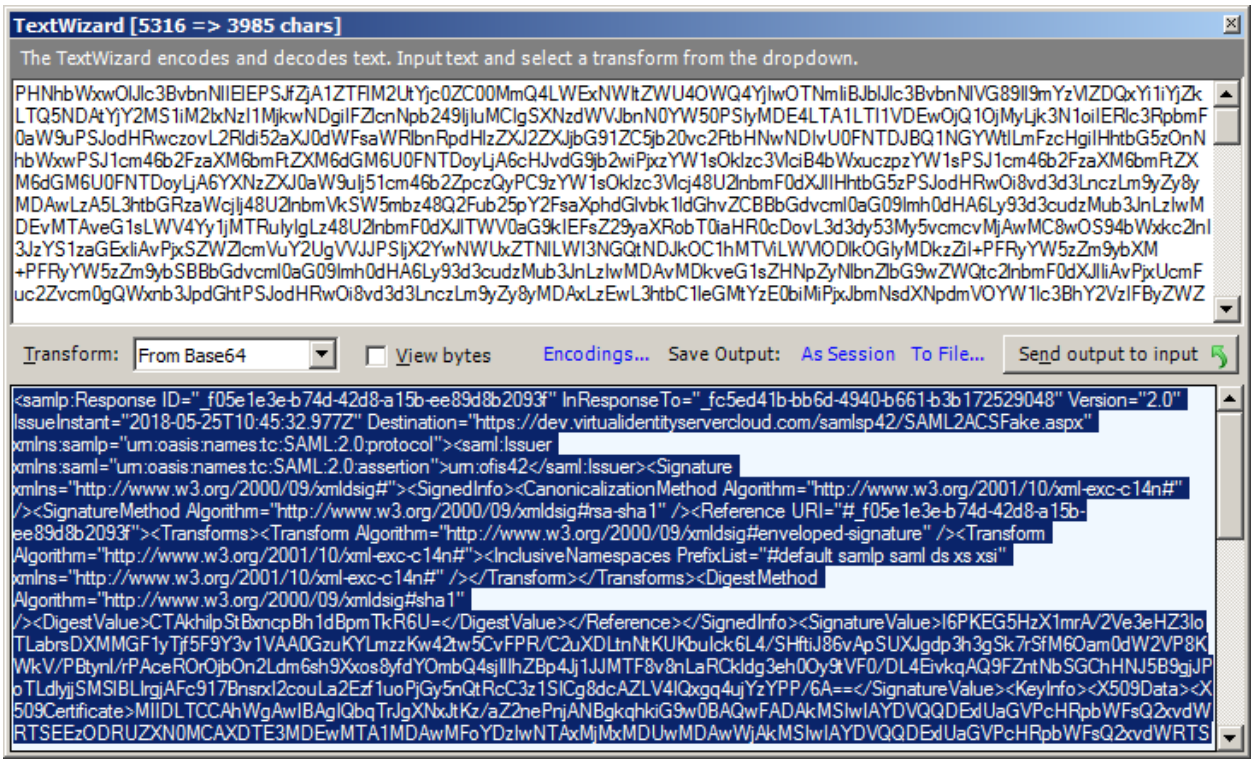
Error on authentication response to the Relying Party

The SAML 2.0 reponse is an HTTP POST request with the following form data. You can see the form data by selecting the line in the request list and then going to the Inspectors -> Web Forms tab. The form data for the SAML 2.0 authentication response are:

- SAMLResponse – encoded SAML 2.0 response

The SAML2Response is base64 encoded. Fiddler can easily decode this for you and show you the XML document. Simply right click on the SAMLResponse value and select “Send to TextWizard

…” That will bring up the Fiddler TextWizard window. If you don’t see XML make sure the Transform: drop down in the middle is set to “From Base64”.



From the SAML 2.0 response XML validate the following:

- Make sure the IdP URI matches the value configured on the relying party. This value can be found in the SAML 2.0 response XML Issuer element.
- Make sure the signing certificate matches the signing certificate configured on the relying party. The signing certificate can found in the Response response XML in the Certificate element under the Signature element.
- Make sure the assertion audience matches the relying party URI. The assertion audience can be found in the Audience element of the assertion.

Diagnosing Common OAuth2\OpenID Connect Federation Issues

OAuth2 and OpenID Connect define different Grant types. Depending on the Grant type the flow may consist of a mixture of web application and web service (REST) calls. The most commonly used Grant is the Authorization Code grant. In this Grant the users browser is used to make a web application authentication request after which an Authorization Code is returned to the web application. The web application makes a REST call to the IdP to exchange the authorization code for a Access Token and JSON Web Token (Jwt).

Error on Authorization Code Grant request to the Identity Provider

If in the Authorization Code Grant request you get an error on the Identity Provider, run a Fiddler trace reproducing the issue. Then look for a GET request to the IdP with the following URL parameters shown below. You can see the URL parameters by selecting the line in the request list and then going to the Inspectors -> Web Forms tab. The URL parameters for the OAuth2\OpenID Connect authentication request are:

- response_type = code
- client_id = <relying party URI>
- redirect_uri = <URL where the authorization code should be returned to>
- scope = <requested authorization>
- state = <federation context>
- nonce = <random value>

Check the following:

- Make sure the client_id value matches the relying party URI configured in the IdP.
- Make sure the redirect_uri value matches what is configured for the relying party in the IdP.

Error getting the Access Token or Jwt using the Authorization Code

The REST call to exchange the Authorization Code for an Access Token and/or Jwt is performed by the relying party. To view this exchange you must run Fiddler on the server that is performing the REST call.

After capturing the REST call with Fiddler, look for the REST call with the following URL paramters:

- code = Authorization Code
- client_id = <relying party URI>
- redirect_uri = <URL where the authorization code should be returned to>
- grant_tyoe = authorization_code

Check the following:

- Make sure the client_id value matches the relying party URI configured in the IdP.
- Make sure the redirect_uri value matches what is configured for the relying party in the IdP.
- Check that the Authorization Code had not be used before (they may only be used once).
- Check that the Authorization Code is not expired (they are typically short lived).

References:

https://www.telerik.com/download/fiddler

https://social.technet.microsoft.com/wiki/contents/articles/3286.ad-fs-2-0-how-to-use-fiddler-web-debugger-to-analyze-a-ws-federation-passive-sign-in.aspx