Web Service Security troubleshooting

- Help-yourself information for component BC-SEC-WSS - SOAP Web Service Security
- Information resources
  - General
  - SAML Holder of Key with an external Security Token Service
    - AS ABAP as Consumer:
    - AS ABAP as Provider:
- Troubleshooting typical errors
  - No Entity in with ID client <SAP system client> found
  - Signing certificate of the SOAP message is not trusted
  - No trusted certificate for SAML authentication found <certificate DN>
  - No trusted certificate for SAML authentication found in USREXTID for <certificate DN>
  - No SAML mapping data in USREXTID for <mapping key> Certificate <certificate DN>
  - "Unknown signer or recipient" - method RESOLVE_PRIVATE_KEY
  - .Net UsernameToken and the password type problem
  - WS Security Policy versions in the WSDL
  - Application Service Provider and Security Token Service Metadata cannot be downloaded in transaction SAML2
- Typical errors during communication with non-SAP SOAP runtimes
  - SAML authentication with non-SAP SAML assertions - general statement
  - CX_ST_MATCH_ELEMENT Exception - Mostly occurs with 3rd party vendors’ SAML assertions
- CX_WS_SECURITY_FAULT Faultcode InvalidSecurity, Function SSFPSE_PARAMETER
- SSL Client certificate chosen in logical port does not seem to be used in communication

Help-yourself information for component BC-SEC-WSS - SOAP Web Service Security

Dear customer,

SOAP WS Security issues can be complex as the SOAP standards allow a very high degree of freedom.
In order to speed up processing of your SOAP Web Service Security problem and to avoid further re-inquiries please provide SAP with all the information listed below.

1. What is the security scenario you want to achieve?
   1. Which authentication method (Username/PW, X.509, SAML)
   2. which transport security (TLS, Message Signature)
   3. Attach guides/links you are using in order to configure your scenario
2. What is the consumer (client) system SOAP runtime and version, e.g. AS ABAP SAP_BASIS release xx?
3. What is the provider (server) system SOAP runtime and version?
4. If you’re using SAP PI. Which Adapter(s) is/are involved on consumer/provider side?
5. Which security settings are configured on consumer and provider side?
   1. Describe by text e.g. "SSL Client certificate authentication" or provide screenshots of logical port security settings and provider binding.
   3. When using SAML, describe which SAML version and mechanism is in use.
      1. AS ABAP supports version SAML 1.1 or SAML 2.0 with the mechanisms
      2. SAML sender-vouches (SAML 1.1 only) or SAML holder-of-key.
6. What's the error message on provider side?
7. What's the error message on consumer side?
8. Provide payloads and communication traces according to the SAP products used in your scenario and your problem.

Provide the payloads as standard txt or xml files, no word, screenshots or PDFs. Please also use exactly the tools listed below to provide the traces/payloads

AS ABAP - In any case provide an SRT_UTIL Payload Trace:
see http://help.sap.com/saphelp_nw70ehp2/helpdata/en/8d/3ceaa36558452e852656b0371e7a49/content.htm?frameset=en/8d/3cea a36558452e852656b0371e7a49/frameset.htm&current_toc=en/d/802cf0454211d189710000e8322d00/plain.htm&node_id=557
If transaction SRT_UTIL of your release does not provide this feature you can use the SOAMANAGER to perform a payload trace as well.
see https://help.sap.com/saphelp_nw70ehp1/helpdata/en/46/ef607dd03d0e65e1000000a155369/content.htm

AS ABAP - Provide an ICM trace in case you received an ICM_HTTP_SSL_ERROR or you think the problem is related to an SSL problem
see http://wiki.scn.sap.com/wiki/x/c4E_FQ

AS ABAP - Provide a kernel trace if you make use of transport authentication like SAP Logon Tickets, X.509 SSL client certificates or HTTP basic authentication
see report documentation of report SEC_TRACE_ANALYZER on how to conveniently provide a kernel trace. More information
is be found in the following note 2181120 (http://service.sap.com/sap/support/notes/1688545).

AS ABAP - When making use of message signatures like in scenarios with SAML or XML Signature/Encryption also a kernel trace can give more insight on what went wrong. Provide a kernel trace according to note 2181120 (http://service.sap.com/sap/support/notes/1688545)

PI on AS Java - Provide XPI Inspector traces see note 1514898 (http://service.sap.com/sap/support/notes/1514898)
Relevant Examples are SOAP Adapter or Custom example with components com.sap.security.core.ws.wss.WSSProcessorBean and com.sap.aia.af.service.administration.monitoring.

Also see the following list of troubleshooting notes. Feel free to use them in order to help yourself.

1. SAP Note 1254821 (http://service.sap.com/sap/support/notes/1254821) - Web Service Security SAML in AS ABAP
2. 1318906 (http://service.sap.com/sap/support/notes/1318906) in case you received an ICM_HTTP_SSL_ERROR

Information resources

General

Furthermore here are relevant links to the official SAP information resources about Web Services Security.

1. Authentication for Web Services - an Introduction
   http://help.sap.com/saphelp_nw73ehp1/helpdata/en/a8/8d5ccfda4a49ab9a2c66618da7894d/content.htm
2. Note: SAML for Web Services
   http://service.sap.com/sap/support/notes/1254821
3. Documentation: SAML Token Profile and how to configure it
4. Difference between SAML sender-vouches and SAML holder-of-key
5. Web Service Security interoperability WIKI with various vendors
6. Authentication for Web Services
   http://help.sap.com/saphelp_nw74/helpdata/en/49/9d40514133eee10000000a421937/content.htm?frameset=en/48/ac79f1b8fb3206e10000000a42189c/frameset.htm&current_toc=en/1c/ad1640033ae569e10000000a155106/plain.htm&node_id=35&show_children=false
7. German Book: "Single Sign-On mit SAP" - the last three chapters deal with Web Services Security
   https://www.rheinwerk-verlag.de/single-sign-on-mit-sap_2409/

SAML Holder of Key with an external Security Token Service


AS ABAP as Consumer:

SOAMANAGER-Setup includes

- Creating a connection to the STS via WS Metadata Exchange
- Configuring the logical port to call the Web Service via SAML

Configuring SSO/STS Scenario SAML Holder-of-key in the WS Consumer AS ABAP http://tinyurl.com/98zzqyp

AS ABAP as Provider:

1. Trusting the STS/IDP

Preparing the WS Provider AS ABAP for Accepting SAML Token Profiles for Validation with the SAML 2 Infrastructure http://help.sap.com/saphelp_nw73/helpdata/en/65/43d66d425d478aa6daeb1366633b43/content.htm?frameset=en/e0/efe61f938e4ab19471c64b1a2268e4/frameset.htm&current_toc=en/1c/ad1640033ae569e10000000a155106/plain.htm&node_id=250&show_children=false

2. SOAMANAGER-Setup

Configuring SSO/STS Scenario SAML Holder-of-key in the WS Provider AS ABAP http://help.sap.com/saphelp_nw73/helpdata/en/0f/757c9c108d4472b471a176396619/content.htm?frameset=en/49/d91d9d6d42fae10000000a42189d/frameset.htm&current_toc=en/1c/ad1640633ae569e10000000a155106/plain.htm&node_id=261

Troubleshooting typical errors
The following describes typical errors one can see in the error log of transaction SRT_UTIL with the most common solutions.

**No Entity in with ID client <SAP system client> found**

"No Entity in with ID client xxx found" means that no trusted Security Token Service (STS) could be found that matches the issuer value of your SAML assertion. Usually that means you need to make sure to have the issuing system of the SAML assertion added as trusted STS configured in transaction SAML2. Note 1254821 - SAML authentication for Web services in AS ABAP (http://service.sap.com/sap/support/notes/1254821) describes the configuration procedure in section 2.2 Mapping using the SAML2 framework. SAP recommends to make use of the metadata based configuration approach in order to avoid human errors.

In some cases however the configuration has already been done but still the message above appears. In this case you should extract the SAML issuer string out of the SAML assertion and compare it with the name of the trusted Security Token Service in transaction SAML2. While comparing those two values also consider white spaces.

SAML 2.0 example assertion containing issuer "https://mycompany.com/STS":

```xml
<Assertion ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc" IssueInstant="2003-04-17T00:46:02Z" Version="2.0" xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
  <Issuer>https://mycompany.com/STS</Issuer>
  ...
</Assertion>
```

SAML 1.1 example assertion containing issuer "https://mycompany.com/STS":

```xml
  <Issuer>https://mycompany.com/STS</Issuer>
  ...
</Assertion>
```

**Signing certificate of the SOAP message is not trusted**

The certificate used to sign the message is not trusted. That means it has not been imported into the PSE that is used for checking the signature. Along with the error message you also the PSE that is used to verify the signature. Importing the signing certificate into this PSE will establish the trust.

For instance you see /usr/sap/<SAP System ID>/DVEBMGS<inst no>/sec/SAPWSSE_WSSKEY.pse as profile. That means you need to import the certificate of the private key used to sign into the "WS Security Keys PSE" using transaction STRUST.

**No trusted certificate for SAML authentication found <certificate DN>**

The certificate for signing the message has not been imported into the System PSE.

**No trusted certificate for SAML authentication found in USREXTID for <certificate DN>**

No mapping entry with that certificate was found in table USREXTID. If this is unexpected, check if exactly the certificate DN as used in the error message is included in your USREXTID mappings.

**No SAML mapping data in USREXTID for <mapping key> Certificate <certificate DN>**

No correct mapping entry with the mapping key and the certificateDN was found in USREXTID. Make sure the certificate DN as used in the error message and the mapping key as in the error message is included in your USREXTID mappings.

"Unknown signer or recipient" - method RESOLVE_PRIVATE_KEY
In order to DECRYPT the SOAP message which was encrypted by the consumer with a certificate of a PSE residing in the provider system the provider system looked up all the private keys. As no key couldn't be found which can be used to DECRYPT the SOAP message an error is thrown.

Make sure on consumer side to use an encryption certificate that was exported from one of the Web Service Security PSEs of the provider system.

Documentation Links: Configuring Systems for the Use of Encryption

.Net UsernameToken and the password type problem

.Net errors when a consumer sends a Password in a UsernameToken that has a type specified, e.g. `<wsse:Password Type="http://docs.oasis-open.org/wss/2004.01/oasis-200401-wss-username-token-profile-1.0#PasswordText">password</wsse:Password>`. This is against the standard but Microsoft didn't fix this in the past. Therefore SAP AS ABAP does not send the type anymore. Unfortunately there are other clients that demand this type as obligatory which is against the standard as well.

The Username Token profile standard states that the type attribute is an optional attribute

```
<wsse:UsernameToken
<wsse:Password @Type
```

This *optional* URI attribute specifies the type of password being provided. The table 241 below identifies the pre-defined types (note that the URI fragments are relative to the URI

```
242 for this specification).
```

Link to the standard: [http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-UsernameTokenProfile.pdf](http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-UsernameTokenProfile.pdf)


WS Security Policy versions in the WSDL

AS ABAP supports Security Policy version 1.1 or 1.2 which have xml namespace [http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702](http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702). Namespace [http://schemas.xmlsoap.org/ws/2005/07/securitypolicy](http://schemas.xmlsoap.org/ws/2005/07/securitypolicy) is of an older WS Security Policy version which is not supported. Security Policies defining such a namespace are ignored and WSDL based logical port setup will not work properly.

Typically SOAP vendors support various versions of WS Security Policies. For .Net WCF for example you can define the Security Policy version to be used as follows.

1. Make sure you are using a CustomBinding for your service.
2. Navigate to the security section of the binding.

Here's how the configuration XML could look like:

```
<binding name="NewBinding3">
  <textMessageEncoding />
  <security authenticationMode="UserNameForCertificate"
    messageSecurityVersion="WSSecurity11WSTrust13WSSecureConversation13WSSecurityPolicy12">
    <secureConversationBootstrap />
  </security>
  <httpTransport />
</binding>
```

Application Service Provider and Security Token Service Metadata cannot be downloaded in transaction SAML2

Going to transaction SAML2 clicking Metadata. In the opening dialog the checkboxes “Application Service Provider” and “Security Token Service Metadata” are greyed out.
Solution:

1. Execute Report WSS_SETUP.
2. Make sure the combobox behind SAML 1.1 Trust is set to “Use SAML Trust”.
3. Uncheck “Test Run”.
4. Press F8 to execute the report.

Now the metadata download options should be visible again in transaction SAML2.

Typical errors during communication with non-SAP SOAP runtimes

The SAP ABAP SOAP Security runtime took part in several Web Service interoperability testing initiatives. However between the vendors still different interpretations of the WS* Standards exist. This results in different implementations resulting in different SOAP xml files. Furthermore the WS* Standards especially in terms of security allow a wide range of mechanisms. SAML as authentication method for instance allows sender-vouches or holder-of-key as method of attesting a user’s identity.

1. Therefore the first step to get to an interoperable setup is to review the documentations of both systems and find out how to configure the desired security mechanism (e.g. SAML sender-vouches). Make sure there is a common denominator between the security mechanisms offered in the documentation. Otherwise this means your scenario is not supported.
2. Still it could be the case that both systems support the mechanism but they are implemented in a different way. This will result in error messages during processing. Typical error messages for AS ABAP are CX_ST_MATCH_ELEMENT or part not signed (e.g. part not signed: wsse:Security/saml:Assertion). In order to troubleshoot these errors follow the advices in the box below.

Troubleshooting interoperability problems between SAP and 3rd party systems

See Single sign on for Web Services for an overview on how to configure interoperability between prominent SOAP runtimes and the SAP Web AS ABAP. If your scenario is not listed we recommend the following the approach:

To see if a certain security mechanism is supported in the same way by two SOAP runtimes a practical approach is to compare the SOAP messages that are produced.

E.g. you want to leverage Single Sign-On with SAML sender-vouches between a 3rd party vendor consumer and AS ABAP as provider. Therefore you configure the desired scenario in both systems independently where the systems are communicating with themselves.

- One time SAP is consumer and provider
- the other time the 3rd party runtime is consumer and provider.
- By intercepting and comparing the SOAP messages you get in the communication you change the configuration of both systems until the messages are similar.
  - Make sure
    - the same elements in the same occurrence exist in the SOAP security header (e.g. a <Signature> element or the <Assertion> element which is the SAML assertion).
    - when having a <Signature> element make sure the <SignedInfo> element contains the same <References> in both messages. Essentially the error “part not signed: wsse:Security/saml:Assertion” indicates that AS ABAP expected the <Assertion> element to be signed but it was not.
    - In some implementations also the order of these elements play an important role.
  - Afterwards you configure the 3rd party system to communicate with the AS ABAP as provider.

SAML authentication with non-SAP SAML assertions - general statement

Although there is a common SAML standard all the vendors adhere to there are many options to construct an assertion. Unsupported SAML assertions sent to the AS ABAP will result in an exception (for instance CX_SXML_PARSE_ERROR in method CL_HTTP_SAML20->VALIDATE_ASSERTION).

Therefore we released a list of example assertions and supported options in the different AS ABAP versions. Make sure the assertion you send to the SAP system adheres to these examples.

Please refer to SAP Note 1254821 - http://service.sap.com/sap/support/notes/1254821. In the two attachments SAML_Sender_Vouches.pdf and SAML_Holder_of_Key.zip you find this set of supported assertions.

CX_ST_MATCH_ELEMENT Exception - Mostly occurs with 3rd party vendors' SAML assertions

This exception is thrown when parsing the XML SOAP message meaning that the XML sent contains elements that are unexpected by the AS ABAP. The exception will always contain an information about the XML element where the problem occured. Most of the times this error occurs in the SAML scenario when a non SAP system is sending a SAML assertion the AS ABAP does not understand.

An example.
Error Message of AS ABAP: CX_ST_MATCH_ELEMENT : System expected end of element '{urn:oasis:names:tc:SAML:1.0:assertion}SubjectConfirmation'

Explanation: AS ABAP didn't expect the <saml:SubjectConfirmationData/> after <saml:ConfirmationMethod>. Deleting the empty <saml:SubjectConfirmationData/> will solve the issue.

The SAML Assertion sent:

```
<saml:Assertion AssertionID="WE15c087689758db144222e5b92b1e16a1bd03f6c9e9d" 
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:unspecified">
    <saml:Subject>
      <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.0:assertion#X509SubjectName"
NameQualifier="mysts">serialNumber=2009351172435441+cn=TESTER,ou=PERSONNEL,ou=TestOU,o=OU,c=ca</saml:NameIdentifier>
      <saml:SubjectConfirmation>
        <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-vouches</saml:ConfirmationMethod>
      </saml:SubjectConfirmation>
    </saml:Subject>
  </saml:AuthenticationStatement>
</saml:Assertion>
```

AS ABAP expected a SAML assertion as described in attachment SAML_Sender_Vouches.pdf of note 1254821.

```
<saml:Assertion MajorVersion="1" MinorVersion="1" AssertionID="saml-0018FE864EE1DDE86C5371CA2C53C43"
Issuer="BXI/000" IssueInstant="2009-03-26T17:15:36Z" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
  <saml:Conditions NotBefore="2009-03-26T17:15:36Z" NotOnOrAfter="2009-03-26T17:20:36Z"/>
  <saml:AuthenticationStatement AuthenticationInstant="2009-03-26T17:15:36Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:unspecified">
    <saml:Subject>
      <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">DEBOER</saml:NameIdentifier>
      <saml:SubjectConfirmation>
        <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-vouches</saml:ConfirmationMethod>
      </saml:SubjectConfirmation>
    </saml:Subject>
  </saml:AuthenticationStatement>
</saml:Assertion>
```


**CX_WS_SECURITY_FAULT Faultcode InvalidSecurity, Function SSFPSE_PARAMETER**

If this error occurs on provider side most likely the PSE file on the file system used to perform a cryptographic operation like decryption or signature check could not be accessed.

To make sure all PSEs can be accessed correctly open transaction STRUST. Enter edit mode using (CTRL+F1) and expand the tree nodes of all PSEs beginning with "WS Security". All child nodes should have a green symbol. If you see a red symbol there most probably was a problem during the distribution of these PSEs on the file system. Right click on the "WS Security" node and select "Distribute" to trigger the distribution.
In order to find out which PSE exactly should be used during communication, perform a user based kernel by using the SEC_TRACE_ANALYZER tool which is documented in SAP Note 2181120. In the trace file you will see which PSE on the file system was attempted to be accessed by the AS ABAP (e.g. PSE file SAPWSSE.pse maps to "WS Security Keys" node in transaction STRUST).

SSL Client certificate chosen in logical port does not seem to be used in communication

By specifying a certain SSL client PSE in a logical port the security runtime will present the selected certificate to an SSL server during handshake in order to authenticate the client. However there are cases where the setting gets overruled and no certificate is sent. In this cases the SSL Client PSE ANONYM including its trusted certificate list will be used.

1. If the SSL server does not request a client certificate AS ABAP will not use the selected SSL Client PSE of the Logical Port
   1. Make sure the server is configured to request/require an SSL client certificate
2. If the list of accepted Certificate Authority (CA) certificates sent by the SSL server during handshake does not include the CA of the certificate selected in the logical port.
   1. Either import the CA certificate (or self-signed-certificate) of the selected client certificate into the server's certificate store
   2. or sign the chosen client certificate by a trusted CA of the ssl server.