# Test Plan
# Security Assertion Markup Language Protocol Interface
# BC-AUTH-SAML 1.0

**SAP WebAS 6.40**

Version 1.0

## Copyright

# 1 Test plan SAML Interoperability (BC-AUTH-SAML 1.0)

## 1.1 Test overview

The BC-AUTH-SAML 1.0 certification tests the interoperability of a third party SAML (Security Assertion Markup Language) source site (see SAML specification at http://www.oasis-open.org/specs/index.php#samlv1.1) with the SAP Web Application Server (WebAS) Java 640 as of Stack 9 or the SAP Enterprise Portal 6.0 as of Stack 9.

The WebAS supports the versions 1.0 and 1.1 of the SAML Browser/Artifact profile for authentication to web resources.

Objective of the test is the demonstration that a resource hosted on the WebAS can be visualized in a web browser (Microsoft Internet Explorer or Mozilla Firefox) without explicit authentication to the WebAS, using the SAML source site implementation of the certified product.

The realization of the visualization (for example a portal application) and the steps needed to authenticate to the visualization component (if required) are not stipulated.

If the vendor only offers a programmatic API to the SAML source site function and is unable to bring in an external portal application that makes use of this implementation, the vendor may create a servlet running in an external web container that uses the API to create a SAML artifact using some authentication data and send a HTTP redirection to a web browser that points to the resource on the J2EE Engine.

## 1.2 Test environment

The BC-AUTH-SAML 1.0 certification procedure requires the following preconfigured hardware and software.

Provided by SAP:

- A current SAP WebAS of release 6.40 Stack 9 or above including a Java stack with the Java Cryptographic toolkit (required for SSL) being installed.
- A client system with Microsoft Internet Explorer 6.0 or later and Mozilla Firefox 1.0 or later being installed.

Provided by vendor:

- The third party SAML source software including required hardware.
- Any installation and configuration tools required as part of the security product’s infrastructure.

# 2 Preparations for the test day

The third party vendor has to provide in advance documentation on how to integrate the SAP engine into the SSO environment of his product.

Documentation about the setup and use of SAML in the WebAS can be found at the SAP help portal at

http://help.sap.com/saphelp_nw04/helpdata/en/94/695b3ebd564644e10000000a114084/frameset.htm

where also information about the SAP provided test application can be found, which is used during the tests

http://help.sap.com/saphelp_nw04/helpdata/en/07/1cf2f1c3227d439c6481127952c8f3/frameset.htm

This documentation also lists the constraints of the SAP implementation (e.g. no support for digital signatures). Some of the listed limitations are removed according to SAP Note 794794.

SAP Note 741157 provides further details regarding the SAML implementation in the WebAS.

## 2.1 Testing of the application before the certification day

Before scheduling the certification day, the vendor is required to demonstrate beforehand that the main use case (see 3.3) can be achieved.

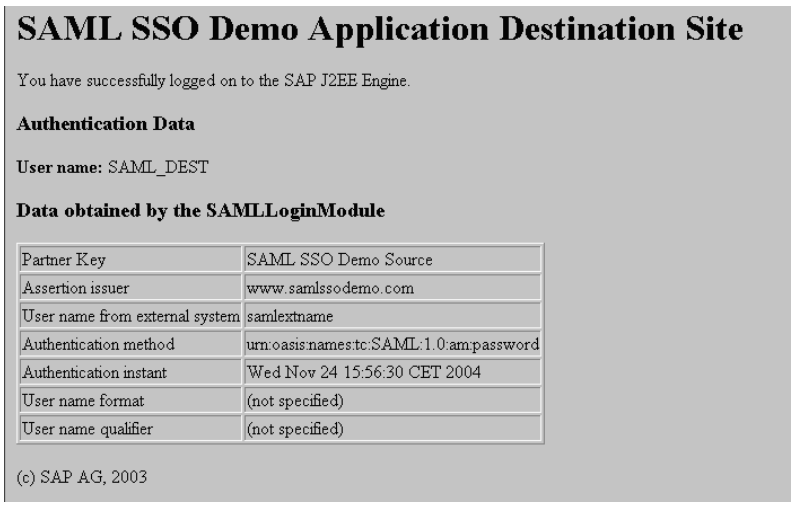
This requires that the vendor installs and configures a SAP WebAS Java of the required version and performs the configuration steps needed to make the WebAS Java accept SAML artifacts for authentication.

A screen-capture showing the web resource

/samlssodemo_dest/destination

of the WebAS Java demonstrates the success.

This web resource is part of the SAML test Application delivered with the WebAS Java which is able to determine that the authentication was performed using SAML Browser/Artifact profile.



**SAML SSO Demo Application Destination Site**

You have successfully logged on to the SAP J2EE Engine.

**Authentication Data**

**User name:** SAML_DEST

**Data obtained by the SAMLLoginModule**

| | |
|---|---|
| Partner Key | SAML SSO Demo Source |
| Assertion issuer | www.samlssodemo.com |
| User name from external system | samlextname |
| Authentication method | urn:oasis:names:tc:SAML:1.0:am:password |
| Authentication instant | Wed Nov 24 15:56:30 CET 2004 |
| User name format | (not specified) |
| User name qualifier | (not specified) |

(c) SAP AG, 2003

## 2.2 Setup options

The connection between the SAML destination service and the SAML source site (assertion

query for artifact) needs to be protected by SSL. If the SAML source site does not support SSL protected connections for the assertion retrieval callback channel, the certificate will not be issued.

For convenience in local testing without Java Cryptographic toolkit, it is possible to disable the enforcement of SSL in the WebAS Java for the SAML Browser/Artifact profile by setting the option "saml → Configuration → Settings → PermitInsecureConnections" to "true" using the Configuration Adapter in the Visual Administrator.

To authenticate the assertion retrieval, the WebAS supports basic authentication as well as client certificate. The supported authentication methods of the source site will be documented in the certification document. Each supported authentication method will be tested during certification and the result for each of them is documented. If none of the authentication methods works successfully, the certificate will not be issued.

The WebAS in the standard delivery does not provide means for a user mapping between the externally provided user name (element "NameIdentifier" in the SAML assertion) and the internal user names.

If possible, the SAML source site should offer user name mapping. If this function is available, its usage will be part of the test.

If the source site implementation does not offer user mapping, this will be documented in the certification document. In this case a workaround in the WebAS will be used that performs such user mapping. This workaround is not available to customers. Customers will then need to create own mapping code in the WebAS. This fact will be stated in the certification document.

# 3 The test procedure

## 3.1 Setup of the SAP J2EE Engine

The following description is an extract of the documentation of the WebAS Java, describing the steps needed to prepare the SAML test application to be accessible with SAML.

- Start the service “SAML” on all server nodes.
- Start the application “sap.com/tc~sec~app” on all server nodes.
- Log on to resource “/samlssodemo_source/setup” and enter the following data:
    - **Configure as source site:** Remove checkbox
    - **Configure as destination site:** Set checkbox
    - **Key:** “SAMLCertification”
    - **Source ID (Hex):** Enter the source ID that the source site implementation will use to identify itself in the artifact. Enter the data as 20 bytes in hexadecimal notation (no gaps; letters can be entered in either upper or lower case style).
    - **Responder Destination Name:** “SAMLCertification”
    - **Responder URL:** Enter the URL of the SAML responder service of the source site implementation to which the request for the assertion shall be sent
    - **User for site:** Leave the proposal “SAML_DEST” / “abcd1234”
    - **User for responder:** If the authentication to the responder service of the source site supports basic authentication, enter the required logon data into these fields. If only client certificate is supported, leave the proposal “SAML_RESP” / “abcd1234”. In a later step, the client certificate will be configured.
    - **SAML without mapping module:** Set this radio button if the SAML source site supports user mapping and can fill the “NameIdentifier” element of the SAML assertion with a freely configurable username. For this test, the source site must issue the assertions with NameIdentifier SAML_DEST.
    - **SAML with mapping module:** Set this radio button if the SAML source site does not support user mapping and fills the “NameIdentifier” element of the SAML assertion with a fixed attribute from its own user repository (e.g. with a DN from a directory). In this case, configure a mapping from the external username (as appearing in the NameIdentifier element) to the internal user SAML_DEST.
      If this option is chosen, then *before* submitting the web page further steps are required in the Visual Administrator Tool of the WebAS:
        - In service “Security Provider”, navigate to “Runtime → User management → Manage Security Stores”.
        - Button “Add Login Module”
        - In dialog “Choose editor for login module options” leave defaults and press OK.
        - Class Name: com.sap.security.core.server.saml.app.ssotest.dest.SAMLMappingModule
        - Display Name: SAMLMappingModule

        - Description: SAML mapping module
        - Select all lines in table “Authentication Mechanisms” and press the “Add” button for “Suitable Authentication Mechanisms”.
        - OK
    - **Basic authentication as fallback:** Set checkbox.
- Submit the web form by pressing “Configure”. In the resulting protocol no errors should be noted.

## 3.2 Access to resource without SAML

To verify that the resource is correctly configured for basic authentication as fallback, access the resource

/samlssodemo_dest/destination

with the web browser. A logon popup must appear. After logging on as SAML_DEST with password “abcd1234”, the resource is shown, but a warning appears that the resource was not accessed using SAML.

## 3.3 Access to resource with SAML using ArtifactReceiver

Authenticate to the SAML source site and perform the steps needed there to make the web browser visualize the resource

/samlssodemo_dest/destination

hosted by the WebAS Java.

In this test, the source site must make the web browser to sent a HTTP GET request to the WebAS Java targeting the resource

/saml/receiver

with the two parameters “SAMLart” (for the artifact) and “TARGET” (for the resource to be shown finally, in this case the “/samlssodemo_dest/destination”).

The test is successful if the resource appears in the web browser without needing to authenticate the user SAML_DEST. In the table showing the data extracted by the SAML login module, the user name contained in the artifact can be seen. If the source site performed the user mapping, here the name SAML_DEST will appear, otherwise the external username will appear.

## 3.4 Access to resource with SAML without ArtifactReceiver (optional)

In this test, the source site must make the web browser to sent a HTTP GET request to the WebAS Java targeting the resource

/samlssodemo_dest/destination

with the parameter “SAMLart” (for the artifact).

The test is successful if the resource appears in the web browser without needing to authenticate the user SAML_DEST.

The ability of the source site to let the web browser access the target resource directly (without using an ArtifactReceiver) will be noted in the certification document.

## 3.5 Check of one-time behavior of source site for artifact query

According to the SAML specification, an artifact that was once resolved to an assertion cannot be used to repeat this resolution.

In order to verify that the source site implements this security relevant part of the specification, perform the test as described in 3.3 and extract the GET request that was finally sent to the WebAS Java.

If the resource appears exclusively in the browser window this can be done by copying the URL from the URL input field.

If the resource appears embedded in a portal application this can be done by viewing the page source and extracting the URL from there.

Remove all session information from that URL (e.g. j_sessionid) except the “SAMLart” parameter, then open a new web browser window and let the browser request the modified URL.

Result: The resource must NOT be delivered and you get a logon popup for basic authentication.

In the log file of the WebAS Java (security.<n>.log), you find information that the artifact could not be resolved from the source site. If available, a log file of the source site should contain information that an artifact was requested that does not exist (or was already consumed).

## 3.6 Artifact query using client certificate (optional)

If the source site supports artifact query using client certificate for authentication, configure the callback channel from WebAS Java to the source site for client certificate.

The configuration of the callback channel user is done in the “Destination Service”.

After having changed the authentication method from “Basic” to “Client certificate”, repeat test 3.3.

## 3.7 User mapping from source site (optional)

During the setup, you have decided whether to use “SAML without mapping module” or “SAML with mapping module”, depending on the capabilities of the source site.

In this test the other option will be tested, if supported by the source site. For this, reconfigure the WebAS Java according to the instructions of 3.1 and repeat test 3.3.