



OIO Bootstrap Token Profile

Version 1.1



Document History	3
Introduction	4
Notational Conventions	4
Characteristics of bootstrap tokens	4
Related profiles	5
Assumptions	5
Token Requirements	7
Processing rules	7
Embedding the Bootstrap Token	8
Example	8
References	9

Content Document History

Date	Version	Initials	Changes
09-06-2009	0.95	SPN	Document ready for OIO public hearing
04-09-2009	1.0.0	TG	Document updated after public hearing (only editorial changes)
14-12-2009	1.0.1	TG	Updated introduction to mention other types of bootstrapping scenarios than browsers; no normative requirements changed.
22-01-2017	1.1	TG	<p>References to deprecated Liberty Profiles have been removed and references have been updated.</p> <p>Requirements for the AttributeStatement to contain OCES attribute profile or persistent pseudonyms has been removed.</p> <p>The bootstrap token is now directly embedded in an attribute of the surrounding assertion.</p> <p>Requirement of bootstrap token not to include an <AuthnStatement> has been relaxed.</p> <p>Improved clarity of requirements by usage of RFC 2119.</p>

Introduction

This document defines requirements for bootstrap tokens to be used within Danish eGovernment.

In this profile, a bootstrap token is a special SAML 2.0 assertion that represents a user (contains claims about a user) and which can be exchanged at a specific Security Token Service for other tokens (called identity or access tokens) which in turn provides access to actual services.

In a browser web SSO scenario, a bootstrap token can be embedded in the SAML 2.0 authentication assertion obtained during web browser SSO; see [OIO-SAML-SSO]. The bootstrap token is embedded via a special attribute (called the DiscoveryEPR attribute) in the surrounding authentication assertion. The bootstrap token identifies the user to a WS-Trust Security Token Service (STS), and a Service Provider can exchange the bootstrap token for a new identity / access token which in turn makes it possible to invoke a remote web service on the user's behalf (so-called identity-based web services).

Bootstrapping also takes place in other scenarios for example involving so-called rich clients. Here a rich client may need to invoke foreign services on the user's behalf, and it may use a bootstrap token to contact an STS to get an identity / access token to use for invoking the service. How a rich client obtains a bootstrap token is not specified by the OIOWS profiles as needs vary widely in local deployments.

The reader is assumed to be familiar with the OIOSAML profile.

Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in RFC2119.

The following abbreviations are used:

- Identity Provider (IdP) - a federated authentication service (typically based on SAML 2.0).
- Service Provider (SP) - a web application or portal allowing federated log-in using an Identity Provider.
- Security Token Service (STS) - a service issuing security tokens for web service invocations (typically based on WS-Trust).
- Web Service Consumer (WSC) - an application or client that needs to invoke a foreign identity-based web service in context of a particular user.
- Web Service Provider (WSP) - a provider of an identity-based web service that allows access based on a security token issued by a trusted STS.

Characteristics of bootstrap tokens

The following are characteristics of bootstrap tokens:

- A bootstrap token is issued by an Identity Provider and contains information about user identity (either direct identifiers or pseudonyms) – typically not user access rights (which are placed in identity / access tokens).
- A bootstrap token is used by a Web Service Consumer (WSC) to contact a WS-Trust Security Token Service in order to get an access token issued to an identity-based web service (provided by a Web Service Provider (WSP)).

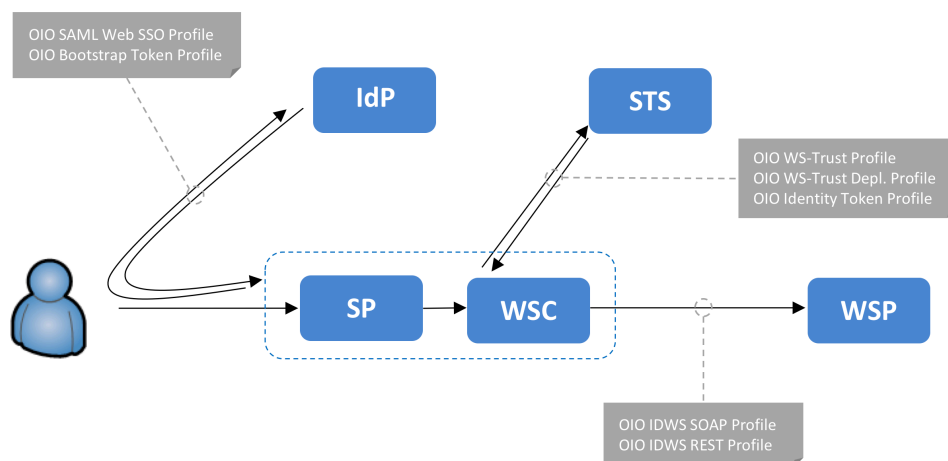
- A bootstrap token can be used as parameter in a WS-Trust call as described in the OIO WS-Trust profile.
- In web SSO scenarios, the bootstrap token is issued by an Identity Provider and is included as an attribute in the SAML authentication assertion issued during web SSO.
- An STS and Identity Provider may have a close relationship (e.g. co-located or part of the same logical system) or be more distributed (e.g. two separate organizations with trust relationships established).
- It is in some scenarios desirable that the STS only issues access tokens if the user has an active session with the IdP. Thus, the STS may contact the IdP to confirm this (via some private protocol) or the STS may rely on the time when the bootstrap token was issued or will expire to establish whether the user had a session recently.

Related profiles

A number of other documents and profiles are closely related:

- The [Scenarios] document describes the overall business goals and requirements and shows how the different OIO profiles are combined to achieve these.
- The OIO Web SSO SAML profile [OIO-SAML-SSO] specifies a SAML 2.0 profile for web SSO. The authentication assertions described in this profile may contain bootstrap tokens defined by this profile. Several elements from the authentication assertion profile are re-used in this profile.

The figure below illustrates the "big picture" of OIO IDWS profiles in a typical scenario:



Assumptions

The profile builds on the following assumptions:

- The Security Token Service receiving a bootstrap token trusts the Identity Provider to assert the user identity.
- If an Identity Provider in a browser scenario issues a SAML SSO assertion during user login, it knows which potential Security Token Services the Services Provider later needs to invoke to obtain access / identity tokens.

- The Identity Provider knows the user identity at the Security Token Services who will be using the bootstrap token.
- It is not a problem that the Web Service Consumer / Service Provider can learn the user ID at the STS from the bootstrap token¹. Since the STS can issue tokens on the user's behalf, the user is assumed to have a trust relationship with the STS.

The above assumptions generally hold true in Danish eGovernment scenarios. In other cases, one would have to introduce a second STS in the architecture to broker trust and map identities; the first STS would be co-located with the IdP and issue tokens for the second STS, which would in turn issue an access token for the desired service.

¹ The IdP cannot easily encrypt the token for the target STS audience since the individual STS will have different public keys. The assumption that one bootstrap token should be used for a several Security Token Services thus implies that encryption of the assertion or name identifiers in the assertion is avoided.

Token Requirements

OIO bootstrap tokens MUST be valid SAML 2.0 assertions conforming to the requirements for OIOSAML assertions for web SSO as defined in [OIO-SAML-SSO] chapter 7 unless explicitly stated otherwise below:

- The token MUST be signed by the issuer.
- The <Conditions> element MUST include one <AudienceRestriction> with the entity IDs of each potential Security Token Service that may later be contacted by the Service Provider for token exchange.
- The bootstrap token SHOULD NOT be encrypted (`saml2:EncryptedAssertion`) or contain encrypted identifiers.
- The bootstrap token SHOULD NOT itself include a Liberty Discovery EPR attribute (the nesting level of tokens should be two at most corresponding to a bootstrap token in an authentication assertion).
- The bootstrap token MAY include private attributes (defined in a separate namespace) that for example identifies the user session at the Identity Provider (e.g. session index). This can be useful if an STS needs to query the Identity Provider to obtain the state of the user's session before new tokens are issued. Such attributes are considered private to IdP-STS implementations.
- The life-time of the token MAY be longer than SSO assertions (which are typically only live a few minutes). The expiration policy is left to concrete implementations.

Processing rules

The STS receiving the bootstrap token SHOULD validate it according normal OIOSAML processing rules.

- It should check that it is mentioned in one of the <AudienceRestriction> elements.

Embedding the Bootstrap Token

When a bootstrap token is embedded in a SAML SSO Assertion it SHOULD be done via a special attribute called `urn:liberty:disco:2006-08:DiscoveryEPR` as described in [OIO-SAML-SSO]².

Example

Below is shown a (simplified) example of an attribute embedding a bootstrap token:

```
<Attribute Name="urn:liberty:disco:2006-08:DiscoveryEPR"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
  <AttributeValue>
    <!-- Here comes the embedded SAML Assertion / bootstrap token -->
    <saml2:Assertion>
      <saml2:Issuer>http://someidp.com</saml2:Issuer>
      <ds:Signature>...</ds:Signature>
      <saml2:Subject>...</saml2:Subject>

      <saml2:Conditions NotOnOrAfter="2008-08-01T21:42:43Z">
        <saml2:AudienceRestriction>
          <saml2:Audience>https://fabricam.com/sts</saml2:Audience>
        </saml2:AudienceRestriction>
      </saml2:Conditions>
      <saml2:AttributeStatement> ...</saml2:AttributeStatement>
    </saml2:Assertion>
  </AttributeValue>
</Attribute>
```

² This is not relevant in rich client scenarios.

References

<

-
- [SAML-CORE] “Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0”, OASIS Standard, 15 March 2005.
- [OIO-SAML-SSO] “OIO Web SSO Profile V2.0”, Danish Digitisation Agency.
- [OIO-IDT] “OIO SAML Profile for Identity Tokens V1.1”, Danish Digitisation Agency.
- [Scenarios] “Identity-Based Web Services – Scenarios”, Danish Digitisation Agency.
-