OAuth Working Group	B. Campbell
Internet-Draft	Ping Identity
Intended status: Standards Track	C. Mortimore
Expires: March 18, 2013	Salesforce
	September 14, 2012

TOC

SAML 2.0 Bearer Assertion Profiles for OAuth 2.0 draft-ietf-oauth-saml2-bearer-14

Abstract

This specification defines the use of a SAML 2.0 Bearer Assertion as a means for requesting an OAuth 2.0 access token as well as for use as a means of client authentication.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as “work in progress.”

This Internet-Draft will expire on March 18, 2013.

Table of Contents

TOC

1. Introduction

The **Security Assertion Markup Language (SAML) 2.0** [OASIS.saml-core-2.0-os] is an XML-based framework that allows identity and security information to be shared across security domains. The SAML specification, while primarily targeted at providing cross domain Web browser single sign-on, was also designed to be modular and extensible to facilitate use in other contexts.

The Assertion, an XML security token, is a fundamental construct of SAML that is often adopted for use in other protocols and specifications. An Assertion is generally issued by an identity provider and consumed by a service provider who relies on its content to identify the Assertion's subject for security related purposes.

The OAuth 2.0 Authorization Protocol [I-D.ietf-oauth-v2] provides a method for making authenticated HTTP requests to a resource using an access token. Access tokens are issued to third-party clients by an authorization server (AS) with the (sometimes implicit) approval of the resource owner. In OAuth, an authorization grant is an abstract term used to describe intermediate credentials that represent the resource owner authorization. An authorization grant is used by the client to obtain an access token. Several authorization grant types are defined to support a wide range of client types and user experiences. OAuth also allows for the definition of new extension grant types to support additional clients or to provide a bridge between OAuth and other trust frameworks. Finally, OAuth allows the definition of additional authentication mechanisms to be used by clients when interacting with the authorization server.

The **OAuth 2.0 Assertion Profile** [I-D.ietf-oauth-assertions] is an abstract extension to OAuth 2.0 that provides a general framework for the use of Assertions as client credentials and/or authorization grants with OAuth 2.0. This specification profiles the **OAuth 2.0 Assertion Profile** [I-D.ietf-oauth-assertions] to define an extension grant type that uses a SAML 2.0 Bearer Assertion to request an OAuth 2.0 access token as well as for use as client credentials. The format and processing rules for the SAML Assertion defined in this specification are intentionally similar, though not identical, to those in the Web Browser SSO Profile defined in **SAML Profiles** [OASIS.saml-profiles-2.0-os]. This specification is reusing, to the extent reasonable, concepts and patterns from that well-established Profile.

This document defines how a SAML Assertion can be used to request an access token when a client wishes to utilize an existing trust relationship, expressed through the semantics of (and digital signature calculated over) the SAML Assertion, without a direct user approval step at the authorization server. It also defines how a SAML Assertion can be used as a client authentication mechanism. The use of an Assertion for client authentication is orthogonal to and separable from using an Assertion as an authorization grant. They can be used either in combination or separately. Client assertion authentication is nothing more than an alternative way for a client to authenticate to the token endpoint and must be used in conjunction with some grant type to form a complete and meaningful protocol request. Assertion authorization grants may be used with or without client authentication or identification. Whether or not client authentication is needed in conjunction with an assertion authorization grant, as well as the supported types of client authentication, are policy decisions at the discretion of the authorization server.

The process by which the client obtains the SAML Assertion, prior to exchanging it with the authorization server or using it for client authentication, is out of scope.

TOC

1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in **RFC 2119** [RFC2119].

Unless otherwise noted, all the protocol parameter names and values are case sensitive.

1.2. Terminology

TOC

All terms are as defined in **The OAuth 2.0 Authorization Protocol** [I-D.ietf-oauth-v2], **OAuth 2.0 Assertion Profile** [I-D.ietf-oauth-assertions], and **Security Assertion Markup Language (SAML) 2.0** [OASIS.saml-core-2.0-os].

---

2. HTTP Parameter Bindings for Transporting Assertions

TOC

The **OAuth 2.0 Assertion Profile** [I-D.ietf-oauth-assertions] defines generic HTTP parameters for transporting Assertions during interactions with a token endpoint. This section defines the values of those parameters for use with SAML 2.0 Bearer Assertions.

2.1. Using SAML Assertions as Authorization Grants

TOC

To use a SAML Bearer Assertion as an authorization grant, use the following parameter values and encodings.

The value of the `grant_type` parameter MUST be `urn:ietf:params:oauth:grant-type:saml2-bearer`.

The value of the `assertion` parameter MUST contain a single SAML 2.0 Assertion. The SAML Assertion XML data MUST be encoded using base64url, where the encoding adheres to the definition in Section 5 of **RFC4648** [RFC4648] and where the padding bits are set to zero. To avoid the need for subsequent encoding steps (by **"application/x-www-form-urlencoded"** [W3C.REC-html401-19991224], for example), the base64url encoded data SHOULD NOT be line wrapped and pad characters ("=") SHOULD NOT be included.

The following non-normative example demonstrates an Access Token Request with an assertion as an authorization grant (with extra line breaks for display purposes only):

```
POST /token.oauth2 HTTP/1.1
Host: as.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Asaml2-bearer&
assertion=PHNhbWxwOl...[omitted for brevity]...ZT4
```

2.2. Using SAML Assertions for Client Authentication

TOC

To use a SAML Bearer Assertion for client authentication grant, use the following parameter values and encodings.

The value of the `client_assertion_type` parameter MUST be `urn:ietf:params:oauth:client-assertion-type:saml2-bearer`.

The value of the `client_assertion` parameter MUST contain a single SAML 2.0 Assertion. The SAML Assertion XML data MUST be encoded using base64url, where the encoding adheres to the definition in Section 5 of **RFC4648** [RFC4648] and where the padding bits are set to zero. To avoid the need for subsequent encoding steps (by **"application/x-www-form-urlencoded"** [W3C.REC-html401-19991224], for example), the base64url encoded

data SHOULD NOT be line wrapped and pad characters ("=") SHOULD NOT be included.

The following non-normative example demonstrates a client authenticating using an assertion during the presentation of an authorization code grant in an Access Token Request (with extra line breaks for display purposes only):

```
POST /token.oauth2 HTTP/1.1
Host: as.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&
code=vAZEIHjQTHuGgaSvyW9hO0RpusLzkvTOww3trZBxZpo&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth
%3Aclient-assertion-type%3Asaml2-bearer&
client_assertion=PHNhbW...[omitted for brevity]...ZT
```

TOC

3. Assertion Format and Processing Requirements

In order to issue an access token response as described in **The OAuth 2.0 Authorization Protocol** [I-D.ietf-oauth-v2] or to rely on an Assertion for client authentication, the authorization server MUST validate the Assertion according to the criteria below. Application of additional restrictions and policy are at the discretion of the authorization server.

- The Assertion's <Issuer> element MUST contain a unique identifier for the entity that issued the Assertion.
- The Assertion MUST contain <Conditions> element with an <AudienceRestriction> element with an <Audience> element containing a URI reference that identifies the authorization server, or the service provider SAML entity of its controlling domain, as an intended audience. The token endpoint URL of the authorization server MAY be used as an acceptable value for an <Audience> element. The authorization server MUST verify that it is an intended audience for the Assertion.
- The Assertion MUST contain a <Subject> element. The subject MAY identify the resource owner for whom the access token is being requested. For client authentication, the Subject MUST be the `client_id` of the OAuth client. When using an Assertion as an authorization grant, the Subject SHOULD identify an authorized accessor for whom the access token is being requested (typically the resource owner, or an authorized delegate). Additional information identifying the subject/principal of the transaction MAY be included in an <AttributeStatement>.
- The Assertion MUST have an expiry that limits the time window during which it can be used. The expiry can be expressed either as the NotOnOrAfter attribute of the <Conditions> element or as the NotOnOrAfter attribute of a suitable <SubjectConfirmationData> element.
- The <Subject> element MUST contain at least one <SubjectConfirmation> element that allows the authorization server to confirm it as a Bearer Assertion. Such a <SubjectConfirmation> element MUST have a Method attribute with a value of `urn:oasis:names:tc:SAML:2.0:cm:bearer`. The <SubjectConfirmation> element MUST contain a <SubjectConfirmationData> element, unless the Assertion has a suitable NotOnOrAfter attribute on the <Conditions> element, in which case the <SubjectConfirmationData> element MAY be omitted. When present, the <SubjectConfirmationData> element MUST have a Recipient attribute with a value indicating the token endpoint URL of the authorization server (or an acceptable alias). The authorization server MUST verify that the value of the Recipient attribute matches the token endpoint URL (or an acceptable alias) to which the Assertion was delivered. The <SubjectConfirmationData> element MUST have a NotOnOrAfter attribute that limits the window during which the Assertion can be confirmed. The <SubjectConfirmationData> element MAY also contain an Address attribute limiting the client address from which the Assertion can be delivered. Verification of the Address is at the discretion of the authorization server.
- The authorization server MUST verify that the NotOnOrAfter instant has not passed, subject to allowable clock skew between systems. An invalid NotOnOrAfter instant on the <Conditions> element invalidates the entire

Assertion. An invalid NotOnOrAfter instant on a <SubjectConfirmationData> element only invalidates the individual <SubjectConfirmation>. The authorization server MAY reject Assertions with a NotOnOrAfter instant that is unreasonably far in the future. The authorization server MAY ensure that Bearer Assertions are not replayed, by maintaining the set of used ID values for the length of time for which the Assertion would be considered valid based on the applicable NotOnOrAfter instant.

- If the Assertion issuer authenticated the subject, the Assertion SHOULD contain a single <AuthnStatement> representing that authentication event.
- If the Assertion was issued with the intention that the presenter act autonomously on behalf of the subject, an <AuthnStatement> SHOULD NOT be included. The presenter SHOULD be identified in the <NameID> or similar element, the <SubjectConfirmation> element, or by other available means like **[OASIS.saml-deleg-cs]**.
- Other statements, in particular <AttributeStatement> elements, MAY be included in the Assertion.
- The Assertion MUST be digitally signed by the issuer and the authorization server MUST verify the signature.
- Encrypted elements MAY appear in place of their plain text counterparts as defined in **[OASIS.saml-core-2.0-os]**.
- The authorization server MUST verify that the Assertion is valid in all other respects per **[OASIS.saml-core-2.0-os]**, such as (but not limited to) evaluating all content within the Conditions element including the NotOnOrAfter and NotBefore attributes, rejecting unknown condition types, etc.

3.1. Authorization Grant Processing

TOC

If present, the authorization server MUST also validate the client credentials.

If the Assertion is not valid, or its subject confirmation requirements cannot be met, the authorization server MUST construct an error response as defined in **OAuth 2.0** [I-D.ietf-oauth-v2]. The value of the `error` parameter MUST be the `invalid_grant` error code. The authorization server MAY include additional information regarding the reasons the Assertion was considered invalid using the `error_description` or `error_uri` parameters.

For example:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store

{
  "error":"invalid_grant",
  "error_description":"Audience validation failed"
}
```

3.2. Client Authentication Processing

TOC

If the client Assertion is not valid, or its subject confirmation requirements cannot be met, the authorization server MUST construct an error response as defined in **OAuth 2.0** [I-D.ietf-oauth-v2]. The value of the `error` parameter MUST be the `invalid_client` error code. The authorization server MAY include additional information regarding the reasons the Assertion was considered invalid using the `error_description` or `error_uri` parameters.

---

4. Authorization Grant Example

TOC

Though non-normative, the following examples illustrate what a conforming Assertion and access token request would look like.

Below is an example SAML 2.0 Assertion (whitespace formatting is for display purposes only):

```
<Assertion IssueInstant="2010-10-01T20:07:34.619Z"
  ID="ef1xsbZxPV2oqjd7HTLRLIBlBb7"
  Version="2.0"
  xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
 <Issuer>https://saml-idp.example.com</Issuer>
 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  [...omitted for brevity...]
 </ds:Signature>
 <Subject>
  <NameID
    Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
   brian@example.com
  </NameID>
  <SubjectConfirmation
    Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
   <SubjectConfirmationData
     NotOnOrAfter="2010-10-01T20:12:34.619Z"
     Recipient="https://authz.example.net/token.oauth2"/>
   </SubjectConfirmation>
  </Subject>
  <Conditions>
    <AudienceRestriction>
      <Audience>https://saml-sp.example.net</Audience>
    </AudienceRestriction>
  </Conditions>
  <AuthnStatement AuthnInstant="2010-10-01T20:07:34.371Z">
    <AuthnContext>
      <AuthnContextClassRef>
        urn:oasis:names:tc:SAML:2.0:ac:classes:X509
      </AuthnContextClassRef>
    </AuthnContext>
  </AuthnStatement>
</Assertion>
```

Figure 1: Example SAML 2.0 Assertion

To present the Assertion shown in the previous example as part of an access token request, for example, the client might make the following HTTPS request (with extra line breaks for display purposes only):

```
POST /token.oauth2 HTTP/1.1
Host: authz.example.net
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Asaml2-
bearer&assertion=PEFzc2VydGlvbiBJc3N1ZUluc3RhbnQ9IjIwMTEtMDU
[...omitted for brevity...]aG5TdGF0ZW1lbnQ-PC9Bc3NlcnRpb24-
```

Figure 2: Example Request

5. Security Considerations

TOC

No additional security considerations apply beyond those described within **The OAuth 2.0 Authorization Protocol** [I-D.ietf-oauth-v2], the **OAuth 2.0 Assertion Profile** [I-D.ietf-oauth-assertions], and in the **Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0**

[OASIS.saml-sec-consider-2.0-os].

6. IANA Considerations

TOC

6.1. Sub-Namespace Registration of urn:ietf:params:oauth:grant-type:saml2-bearer

TOC

This is a request to IANA to please register the value `grant-type:saml2-bearer` in the registry urn:ietf:params:oauth established in **An IETF URN Sub-Namespace for OAuth** [I-D.ietf-oauth-urn-sub-ns].

- URN: urn:ietf:params:oauth:grant-type:saml2-bearer
- Common Name: SAML 2.0 Bearer Assertion Grant Type Profile for OAuth 2.0
- Change controller: IETF
- Specification Document: [[this document]]

6.2. Sub-Namespace Registration of urn:ietf:params:oauth:client-assertion-type:saml2-bearer

TOC

This is a request to IANA to please register the value `client-assertion-type:saml2-bearer` in the registry urn:ietf:params:oauth established in **An IETF URN Sub-Namespace for OAuth** [I-D.ietf-oauth-urn-sub-ns].

- URN: urn:ietf:params:oauth:client-assertion-type:saml2-bearer
- Common Name: SAML 2.0 Bearer Assertion Profile for OAuth 2.0 Client Authentication
- Change controller: IETF
- Specification Document: [[this document]]

7. References

TOC

7.1. Normative References

TOC

[I-D.ietf-oauth-assertions] Campbell, B., Mortimore, C., Jones, M., and Y. Goland, "**Assertion Framework for OAuth 2.0**," draft-ietf-oauth-assertions-05 (work in progress), September 2012 (**TXT**, **PDF**).

[I-D.ietf-oauth-urn-sub-ns] Campbell, B. and H. Tschofenig, "**An IETF URN Sub-Namespace for OAuth**," draft-ietf-oauth-urn-sub-ns-06 (work in progress), July 2012 (**TXT**).

[I-D.ietf-oauth-v2] Hardt, D., "**The OAuth 2.0 Authorization Framework**," draft-ietf-oauth-v2-31 (work in progress), August 2012 (**TXT**, **PDF**).

[OASIS.saml-core-2.0-os] **Cantor, S.**, **Kemp, J.**, **Philpott, R.**, and **E. Maler**, "**Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0**," OASIS Standard saml-core-2.0-os, March 2005.

[RFC2119] **Bradner, S.**, "**Key words for use in RFCs to Indicate Requirement Levels**," BCP 14, RFC 2119, March 1997 (**TXT**, **HTML**, **XML**).

[RFC4648] Josefsson, S., "**The Base16, Base32, and Base64 Data Encodings**," RFC 4648, October 2006 (**TXT**).

7.2. Informative References

TOC

[OASIS.saml-deleg-cs] Cantor, S., Ed., "**SAML V2.0 Condition for Delegation Restriction**," Nov 2009.

[OASIS.saml-profiles-2.0-os] **Hughes, J.**, **Cantor, S.**, **Hodges, J.**, **Hirsch, F.**, **Mishra, P.**, **Philpott, R.**, and **E. Maler**, "**Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0**," OASIS Standard OASIS.saml-profiles-2.0-os, March 2005.

[OASIS.saml-sec-consider-2.0-os] Hirsch, F., Philpott, R., and E. Maler, "Security and Privacy Considerations for the OASIS Security Markup Language (SAML) V2.0," OASIS Standard saml-sec-consider-2.0-os, March 2005.

[W3C.REC-html401-19991224] Hors, A., Raggett, D., and I. Jacobs, "HTML 4.01 Specification," World Wide Web Consortium Recommendation REC-html401-19991224, December 1999 (HTML).

TOC

Appendix A. Acknowledgements

The following people contributed wording and concepts to this document: Paul Madsen, Patrick Harding, Peter Motykowski, Eran Hammer, Peter Saint-Andre, Ian Barnett, Eric Fazendin, Torsten Lodderstedt, Susan Harper, Scott Tomilson, Scott Cantor, Michael B. Jones, Hannes Tschofenig, David Waite, Phil Hunt, and Mukesh Bhatnagar.

TOC

Appendix B. Document History

[[to be removed by RFC editor before publication as an RFC]]

draft-ietf-oauth-saml2-bearer-14

- Add more text to intro explaining that an assertion grant type can be used with or without client authentication/identification and that client assertion authentication is nothing more than an alternative way for a client to authenticate to the token endpoint
- Add examples to Sections 2.1 and 2.2
- Update references

draft-ietf-oauth-saml2-bearer-13

- Update references: oauth-assertions-04, oauth-urn-sub-ns-05, oauth -28
- Changed "Description" to "Specification Document" in both registration requests in IANA Considerations per changes to the template in ietf-oauth-urn-sub-ns(-03)
- Added "(or an acceptable alias)" so that it's in both sentences about Recipient and the token endpoint URL so there's no ambiguity
- Update area and workgroup (now Security and OAuth was Internet and nothing)

draft-ietf-oauth-saml2-bearer-12

- updated reference to draft-ietf-oauth-v2 from -25 to -26 and draft-ietf-oauth-assertions from -02 to -03

draft-ietf-oauth-saml2-bearer-11

- Removed text about limited lifetime access tokens and the SHOULD NOT on issuing refresh tokens. The text was moved to draft-ietf-oauth-assertions-02 and somewhat modified per http://www.ietf.org/mail-archive/web/oauth/current/msg08298.html.
- Fixed typo/missing word per http://www.ietf.org/mail-archive/web/oauth/current/msg08733.html.
- Added Terminology section.

draft-ietf-oauth-saml2-bearer-10

- fix a spelling mistake

draft-ietf-oauth-saml2-bearer-09

- Attempt to address an ambiguity around validation requirements when the Conditions element contain a NotOnOrAfter and SubjectConfirmation/SubjectConfirmationData does too. Basically it needs to have at least one bearer SubjectConfirmation element but that element can omit SubjectConfirmationData, if Conditions has an expiry on it. Otherwise, a valid SubjectConfirmation must have a SubjectConfirmationData with Recipient and NotOnOrAfter. And any SubjectConfirmationData that has those elements needs to have them checked.

- clarified that AudienceRestriction is under Conditions (even though it's implied by schema)
- fix a typo

draft-ietf-oauth-saml2-bearer-08

- fix some typos

draft-ietf-oauth-saml2-bearer-07

- update reference from draft-campbell-oauth-urn-sub-ns to draft-ietf-oauth-urn-sub-ns
- Updated to reference draft-ietf-oauth-v2-20

draft-ietf-oauth-saml2-bearer-06

- Fix three typos NamseID->NameID and (2x) Namspace->Namespace

draft-ietf-oauth-saml2-bearer-05

- Allow for subject confirmation data to be optional when Conditions contain audience and NotOnOrAfter
- Rework most of the spec to profile draft-ietf-oauth-assertions for both authn and authz including (but not limited to):
    - remove requirement for issuer to be urn:oasis:names:tc:SAML:2.0:nameid-format:entity
    - change wording on Subject requirements
- using a MAY, explicitly say that the Audience can be token endpoint URL of the authorization server
- Change title to be more generic (allowing for client authn too)
- added client authentication to the abstract
- register and use urn:ietf:params:oauth:grant-type:saml2-bearer for grant type rather than http://oauth.net/grant_type/saml/2.0/bearer
- register urn:ietf:params:oauth:client-assertion-type:saml2-bearer
- remove scope parameter as it is defined in http://tools.ietf.org/html/draft-ietf-oauth-assertions
- remove assertion param registration because it [should] be in http://tools.ietf.org/html/draft-ietf-oauth-assertions
- fix typo(s) and update/add references

draft-ietf-oauth-saml2-bearer-04

- Changed the grant_type URI from "http://oauth.net/grant_type/assertion/saml/2.0/bearer" to "http://oauth.net/grant_type/saml/2.0/bearer" - dropping the word assertion from the path. Recent versions of draft-ietf-oauth-v2 no longer refer to extension grants using the word assertion so this URI is more reflective of that. It also more closely aligns with the grant type URI in draft-jones-oauth-jwt-bearer-00 which is "http://oauth.net/grant_type/jwt/1.0/bearer".
- Added "case sensitive" to scope definition to align with draft-ietf-oauth-v2-15/16.
- Updated to reference draft-ietf-oauth-v2-16

draft-ietf-oauth-saml2-bearer-03

- Cleanup of some editorial issues.

draft-ietf-oauth-saml2-bearer-02

- Added scope parameter with text copied from draft-ietf-oauth-v2-12 (the reorg of draft-ietf-oauth-v2-12 made it so scope wasn't really inherited by this spec anymore)
- Change definition of the assertion parameter to be more generally applicable per the suggestion near the end of http://www.ietf.org/mail-archive/web/oauth/current/msg05253.html
- Editorial changes based on feedback

draft-ietf-oauth-saml2-bearer-01

- Update spec name when referencing draft-ietf-oauth-v2 (The OAuth 2.0 Protocol Framework -> The OAuth 2.0 Authorization Protocol)

- Update wording in Introduction to talk about extension grant types rather than the assertion grant type which is a term no longer used in OAuth 2.0
- Updated to reference draft-ietf-oauth-v2-12 and denote as work in progress
- Update Parameter Registration Request to use similar terms as draft-ietf-oauth-v2-12 and remove Related information part
- Add some text giving discretion to AS on rejecting assertions with unreasonably long validity window.

draft-ietf-oauth-saml2-bearer-00

- Added Parameter Registration Request for "assertion" to IANA Considerations.
- Changed document name to draft-ietf-oauth-saml2-bearer in anticipation of becoming an OAUTH WG item.
- Attempt to move the entire definition of the 'assertion' parameter into this draft (it will no longer be defined in OAuth 2 Protocol Framework).

draft-campbell-oauth-saml-01

- Updated to reference draft-ietf-oauth-v2-11 and reflect changes from -10 to -11.
- Updated examples.
- Relaxed processing rules to allow for more than one SubjectConfirmation element.
- Removed the 'MUST NOT contain a NotBefore attribute' on SubjectConfirmationData.
- Relaxed wording that ties the subject of the Assertion to the resource owner.
- Added some wording about identifying the client when the subject hasn't directly authenticated including an informative reference to SAML V2.0 Condition for Delegation Restriction.
- Added a few examples to the language about verifying that the Assertion is valid in all other respects.
- Added some wording to the introduction about the similarities to Web SSO in the format and processing rules
- Changed the grant_type (was assertion_type) URI from http://oauth.net/assertion_type/saml/2.0/bearer to http://oauth.net/grant_type/assertion/saml/2.0/bearer
- Changed title to include "Grant Type" in it.
- Editorial updates based on feedback from the WG and others (including capitalization of Assertion when referring to SAML).

draft-campbell-oauth-saml-00

- Initial I-D

Authors' Addresses

TOC

Brian Campbell
Ping Identity Corp.
Email: brian.d.campbell@gmail.com

Chuck Mortimore
Salesforce.com
Email: cmortimore@salesforce.com