B. Campbell, Ed.
Ping Identity Corp.
C. Mortimore
Salesforce.com

Internet-Draft
Intended status: Standards Track
Expires: June 16, 2011

December 13, 2010

SAML 2.0 Bearer Assertion Grant Type Profile for OAuth 2.0 draft-campbell-oauth-saml-01

Abstract

This specification defines the use of a SAML 2.0 bearer Assertion as means for requesting an OAuth 2.0 access token.

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 June 16, 2011.

Table of Contents

1. Introduction

The Security Assertion Markup Language (SAML) 2.0 [OASIS.saml-core-2.0-os], is an XML-based framework that allows for 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.

OAuth 2.0 [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. OAuth defines multiple profiles for obtaining access tokens to support a wide range of client types and user experiences. One such method is one in which the client trades an 'assertion' (not specifically a SAML Assertion) for an access token using the so-called 'assertion grant_type'. However OAuth 2.0 leaves the specific format and validation of the assertion out of scope.

This specification profiles the use of a SAML 2.0 bearer Assertion in requesting an access token using the assertion grant_type from OAuth 2.0. 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 [OASIS.saml-profiles-2.0-os] reusing, to the extent reasonable, concepts and patterns from that well-established profile.

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.

2. SAML Assertion Access Token Request

A SAML Assertion can be used to request an access token when a client wishes to utilize an existing trust relationship, expressed though the semantics of (and digital signature calculated over) the SAML

Assertion, without a direct user approval step at the authorization server.

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

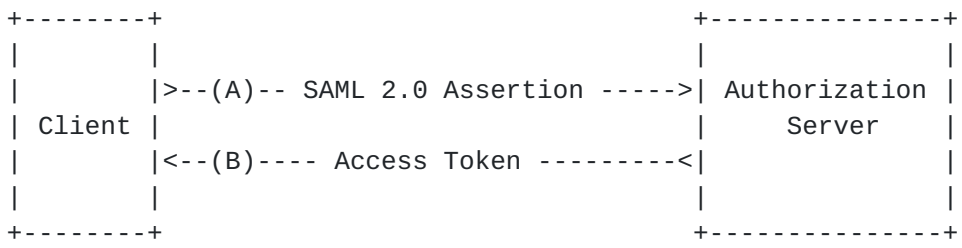

Figure 1: Assertion Access Token Request

The request/response flow illustrated in Figure 1 includes the following steps:

(A) The client sends an access token request to the authorization server with an appropriate OAuth grant_type and includes a SAML 2.0 Assertion.

(B) The authorization server validates the Assertion per the processing rules defined in this specification and issues an access token.

2.1. Client Requests Access Token

The client requests an access token by making an HTTP "POST" request to the token endpoint using SAML Assertion as the access grant. The client makes an access token request, as defined in OAuth, with the following parameter definitions taking precedence in the constructed URI:

grant_type
: REQUIRED. The value of the grant_type parameter MUST be "http://oauth.net/grant_type/assertion/saml/2.0/bearer"

assertion
: REQUIRED. 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 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.

2.2. Assertion Format and Processing Requirements

Prior to issuing an access token response as described in [I-D.ietf.oauth-v2], 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 Format attribute MUST be omitted or have a value of "urn:oasis:names:tc:SAML:2.0:nameid-format:entity".

- The Assertion MUST contain a <Subject> element. The subject MAY identify the resource owner for whom the access token is being requested.

- The <Subject> element MUST contain at least one <SubjectConfirmation> element that allows the authorization server to confirm it as a bearer Assertion. Conditions for bearer subject confirmation are described below.

  * The <SubjectConfirmation> MUST have a Method attribute with a value of "urn:oasis:names:tc:SAML:2.0:cm:bearer" and MUST contain a <SubjectConfirmationData> element.

  * The <SubjectConfirmationData> element MUST have a Recipient attribute with a value indicating the token endpoint URL of the authorization server. 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 authorization server MUST verify that the NotOnOrAfter instant has not passed, subject to allowable clock skew between systems. 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 NotOnOrAfter attribute in the <SubjectConfirmationData>.

    * 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.

- 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 client act autonomously on behalf of the subject, an <AuthnStatement> SHOULD NOT be included. The client 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 contain 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 authorization server MUST verify that it is an intended audience for 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.

2.3. Error Response

If the Assertion is not valid, or its subject confirmation requirements cannot be met, the the authorization server MUST construct an error response as defined in [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"
}
```

2.4. Example (non-normative)

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 2: 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 (line breaks are for display purposes only):

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

grant_type=http%3A%2F%2Foauth.net%2Fgrant_type%2Fassertion%2F
saml%2F2.0%2Fbearer&assertion=PEFzc2VydGlvbiBJc3N1ZUluc3RhbnQ
[...omitted for brevity...]V0aG5TdGF0ZW1lbnQ-PC9Bc3NlcnRpb24-
```

Figure 3: Example Request

3. Security Considerations

No additional considerations beyond those described within the OAuth 2.0 Protocol [I-D.ietf.oauth-v2] and in the Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0 [OASIS.saml-sec-consider-2.0-os].

4. IANA Considerations

This document has no actions for IANA.

Appendix A. Contributors

The following people contributed wording and concepts to this document: Paul Madsen, Patrick Harding, Peter Motyka, Peter Saint-Andre, Ian Barnett, Eric Fazendin, Torsten Lodderstedt, Scott Cantor and David Waite

Appendix B. Document History

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

-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).

-00

- Initial I-D

5. References

5.1. Normative References

[I-D.ietf.oauth-v2]
Hammer-Lahav, E., Ed., Recordon, D., and D. Hardt, "The OAuth 2.0 Protocol Framework", ID draft-ietf-oauth-v2-11, Dec 2010.

[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.

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

5.2. Informative References

[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., Jacobs, I., and D. Raggett, "HTML 4.01 Specification", World Wide Web Consortium Recommendation REC-html401-19991224, December 1999, <http://www.w3.org/TR/1999/REC-html401-19991224>.

Authors' Addresses

Brian Campbell (editor)
Ping Identity Corp.

Email: brian.d.campbell@gmail.com

Chuck Mortimore
Salesforce.com

Email: cmortimore@salesforce.com