

TTA Standard

정보통신단체표준(영문표준)
TTAx.xx-xx.xxxx/R1

제정일: 20xx 년 xx 월 xx 일

공개인증 2.0 클라이언트 인증 및
인가 승인을 위한 SAML 2.0 프로파
일

Security Assertion Markup Language (SAML)
2.0 Profile for OAuth 2.0 Client
Authentication and Authorization Grants

표준초안 검토 위원회 응용보안/평가인증 프로젝트그룹(PG504)

표준안 심의 위원회 정보보호 기술위원회(TC5)

	성명	소속	직위	위원회 및 직위	표준번호
표준(과제) 제안	나재훈	ETRI	책임연구원	PG504 위원장	
표준 초안 작성자	나재훈	ETRI	책임연구원	PG504 위원장	
	강유성	ETRI	책임연구원	PG504 위원	
	이상우	ETRI	책임연구원	PG504 위원	
	나중찬	ETRI	책임연구원		
사무국 담당	김재웅	TTA	단장		
	문서연	TTA	전임연구원		

본 문서에 대한 저작권은 TTA에 있으며, TTA와 사전 협의 없이 이 문서의 전체 또는 일부를 상업적 목적으로 복제 또는 배포해서는 안 됩니다.

본 표준 발간 이전에 접수된 지식재산권 확약서 정보는 본 표준의 '부록(지식재산권 확약서 정보)'에 명시하고 있으며, 이후 접수된 지식재산권 확약서는 TTA 웹사이트에서 확인할 수 있습니다.

본 표준과 관련하여 접수된 확약서 외의 지식재산권이 존재할 수 있습니다.

발행인 : 한국정보통신기술협회 회장

발행처 : 한국정보통신기술협회

13591, 경기도 성남시 분당구 분당로 47

Tel : 031-724-0114, Fax : 031-724-0109

발행일 : 20xx.xx

서 문

1 표준의 목적

이 표준은 클라이언트 인증뿐만 아니라 공개인증(OAuth) 2.0 액세스 토큰을 요청하는 수단으로 SAML (Security Assertion Markup Language) 2.0 베어러 주장의 사용을 정의한다.

2 주요 내용 요약

이 표준은 SAML 2.0 베어러 주장을 사용하여 공개인증 (OAuth) 2.0 액세스 토큰을 요청하고 클라이언트 크리덴셜로 사용하는 확장 승인 타입을 정의하기 위해 공개인증 주장 프레임워크[RFC7521]를 구체화 한다.

이 문서는 사용자가 인가 서버에게 직접적인 승인 단계 없이, 클라이언트가 SAML 주장으로 기존의 신뢰 관계를 활용하고자 할 때 SAML 주장을 사용하여 액세스 토큰을 요청하는 방법을 정의한다. 또한 SAML 주장을 클라이언트 인증 메커니즘으로 사용하는 방법을 정의한다.

3 인용 표준과의 비교

3.1 인용 표준과의 관련성

이 표준은 IETF RFC 7522, “Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grant”을 기반으로 동일하게 작성되었음

3.2 인용 표준과 본 표준의 비교표

TTAK.xx-xx.xxxx/R1	IETF RFC 7522	비고
1~9	-	동일
부록 I -1~6	-	추가

Preface

1 Purpose

This document defines the use of a Security Assertion Markup Language (SAML) 2.0 Bearer Assertion as a means for requesting an OAuth 2.0 access token as well as for client authentication.

2 Summary

This document profiles the OAuth Assertion Framework [RFC7521] 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.

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 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.

3 Relationship to Reference Standards

The standard is fully equivalent to IETF RFC 7522, “Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grant” on May 2015.

목 차

1	소 개	1
1.1	표기법	1
1.2	용어	1
2	주장 전달을 위한 HTTP 파라미터 바인딩	3
2.1	SAML 주장을 인가 승인으로 사용	3
2.2	SAML 주장을 클라이언트 인증으로 사용	4
3	주장 양식과 처리 요구사항	5
3.1	인가 승인 처리	7
3.2	클라이언트 인증 처리	8
4	인가 승인 예	8
5	상호운용성 고려사항	10
6	정보보호 고려사항	10
7	프라이버시 고려사항	13
8	IANA 고려사항	13
8.1	하위 네임스페이스 등록 urn:ietf:params:oauth:grant-type:saml2-bearer	13
8.2	하위 네임스페이스 등록 urn:ietf:params:oauth:client-assertion-type:saml2-bearer	15
9	참고문헌	15
9.1	규범적 참고문헌	15
9.2	정보적 참고문헌	16
부록	I-1 지식재산권 요약서 정보	16
	I-2 시험인증 관련 사항	17
	I-3 본 표준의 연계(family) 표준	18
	I-4 참고 문헌	19
	I-5 영문표준 해설서	20
	I-6 표준의 이력	23

Internet Engineering Task Force (IETF)
Request for Comments: 7522
Category: Standards Track
ISSN: 2070-1721

B. Campbell
Ping Identity
C. Mortimore
Salesforce
M. Jones
Microsoft
May 2015

Security Assertion Markup Language (SAML) 2.0 Profile
for OAuth 2.0 Client Authentication and Authorization Grants

Abstract

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

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7522>.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- 1. Introduction 2
 - 1.1. Notational Conventions 4
 - 1.2. Terminology 4
- 2. HTTP Parameter Bindings for Transporting Assertions 4
 - 2.1. Using SAML Assertions as Authorization Grants 4
 - 2.2. Using SAML Assertions for Client Authentication 5
- 3. Assertion Format and Processing Requirements 6
 - 3.1. Authorization Grant Processing 8
 - 3.2. Client Authentication Processing 9
- 4. Authorization Grant Example 9
- 5. Interoperability Considerations 11
- 6. Security Considerations 11
- 7. Privacy Considerations 12
- 8. IANA Considerations 12
 - 8.1. Sub-Namespace Registration of
urn:ietf:params:oauth:grant-type:saml2-bearer 12
 - 8.2. Sub-Namespace Registration of
urn:ietf:params:oauth:client-assertion-type:saml2-bearer 13
- 9. References 13
 - 9.1. Normative References 13
 - 9.2. Informative References 14
- Acknowledgements 15
- Authors' Addresses 15

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 (SSO), 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. (Some examples include [OASIS.WSS-SAMLTokenProfile] and [OASIS.WS-Fed].) An Assertion is generally issued by an Identity Provider and consumed by a Service Provider that relies on its content to identify the Assertion's subject for security-related purposes.

The OAuth 2.0 Authorization Framework [RFC6749] 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.

"Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants" [RFC7521] 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 Assertion Framework [RFC7521] 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 the SAML Profiles [OASIS.saml-profiles-2.0-os] specification. 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 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 it 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.

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

All terms are as defined in the following specifications: "The OAuth 2.0 Authorization Framework" [RFC6749], the OAuth Assertion Framework [RFC7521], and "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0" [OASIS.saml-core-2.0-os].

2. HTTP Parameter Bindings for Transporting Assertions

The OAuth Assertion Framework [RFC7521] defines generic HTTP parameters for transporting assertions during interactions with a token endpoint. This section defines specific parameters and treatments of those parameters for use with SAML 2.0 Bearer Assertions.

2.1. Using SAML Assertions as Authorization Grants

To use a SAML Bearer Assertion as an authorization grant, the client uses an access token request as defined in Section 4 of the OAuth Assertion Framework [RFC7521] with the following specific parameter values and encodings.

The value of the "grant_type" parameter is "urn:ietf:params:oauth:grant-type:saml2-bearer".

The value of the "assertion" parameter contains a single SAML 2.0 Assertion. It MUST NOT contain more than one 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 RFC 4648 [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 MUST NOT be line wrapped and pad characters ("=") MUST NOT be included.

The "scope" parameter may be used, as defined in the OAuth Assertion Framework [RFC7521], to indicate the requested scope.

Authentication of the client is optional, as described in Section 3.2.1 of OAuth 2.0 [RFC6749] and consequently, the "client_id" is only needed when a form of client authentication that relies on the parameter is used.

The following 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=PHNhbWxwO1...[omitted for brevity]...ZT4
```

2.2. Using SAML Assertions for Client Authentication

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

The value of the "client_assertion_type" parameter is "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 RFC 4648 [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 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=n0esc3NRze7LTCu7iYzS6a5acc3f0ogp4&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Asaml2-bearer&
client_assertion=PHNhbW...[omitted for brevity]...ZT4
```

3. Assertion Format and Processing Requirements

In order to issue an access token response as described in OAuth 2.0 [RFC6749] 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.

1. The Assertion's <Issuer> element **MUST** contain a unique identifier for the entity that issued the Assertion. In the absence of an application profile specifying otherwise, compliant applications **MUST** compare Issuer values using the Simple String Comparison method defined in Section 6.2.1 of RFC 3986 [RFC3986].
2. The Assertion **MUST** contain a <Conditions> element with an <AudienceRestriction> element with an <Audience> element that identifies the authorization server as an intended audience. Section 2.5.1.4 of "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0" [OASIS.saml-core-2.0-os] defines the <AudienceRestriction> and <Audience> elements and, in addition to the URI references discussed there, the token endpoint URL of the authorization server **MAY** be used as a URI that identifies the authorization server as an intended audience. The authorization server **MUST** reject any Assertion that does not contain its own identity as the intended audience. In the absence of an application profile specifying otherwise, compliant applications **MUST** compare the Audience values using the Simple String Comparison method defined in Section 6.2.1 of RFC 3986 [RFC3986]. As noted in Section 5, the precise strings to be used as the Audience for a given authorization server must be configured out of band by the authorization server and the issuer of the Assertion.
3. The Assertion **MUST** contain a <Subject> element identifying the principal that is the subject of the Assertion. Additional information identifying the subject/principal **MAY** be included in an <AttributeStatement>.
 - A. For the authorization grant, the Subject typically identifies an authorized accessor for which the access token is being requested (i.e., the resource owner or an authorized delegate), but in some cases, it may be a pseudonymous identifier or other value denoting an anonymous user.
 - B. For client authentication, the Subject **MUST** be the "client_id" of the OAuth client.

4. 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.
5. The <Subject> element MUST contain at least one <SubjectConfirmation> element that has a Method attribute with a value of "urn:oasis:names:tc:SAML:2.0:cm:bearer". If the Assertion does not have a suitable NotOnOrAfter attribute on the <Conditions> element, the <SubjectConfirmation> element MUST contain a <SubjectConfirmationData> element. 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.
6. The authorization server MUST reject the entire Assertion if the NotOnOrAfter instant on the <Conditions> element has passed (subject to allowable clock skew between systems). The authorization server MUST reject the <SubjectConfirmation> (but MAY still use the rest of the Assertion) if the NotOnOrAfter instant on the <SubjectConfirmationData> has passed (subject to allowable clock skew). Note that 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.
7. If the Assertion issuer directly 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 and the client presenting the Assertion SHOULD be identified in the <NameID> or similar element in the <SubjectConfirmation> element, or by other available means like "SAML V2.0 Condition for Delegation Restriction" [OASIS.saml-deleg-cs].

8. Other statements, in particular <AttributeStatement> elements, MAY be included in the Assertion.
9. The Assertion MUST be digitally signed or have a Message Authentication Code (MAC) applied by the issuer. The authorization server MUST reject Assertions with an invalid signature or MAC.
10. Encrypted elements MAY appear in place of their plaintext counterparts as defined in [OASIS.saml-core-2.0-os].
11. The authorization server MUST reject an Assertion that is not valid in all other respects per [OASIS.saml-core-2.0-os], such as (but not limited to) all content within the Conditions element including the NotOnOrAfter and NotBefore attributes, unknown condition types, etc.

3.1. Authorization Grant Processing

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. However, if client credentials are present in the request, the authorization server MUST validate them.

If the Assertion is not valid (including if its subject confirmation requirements cannot be met), the authorization server constructs an error response as defined in OAuth 2.0 [RFC6749]. 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

If the client Assertion is not valid (including if its subject confirmation requirements cannot be met), the authorization server constructs an error response as defined in OAuth 2.0 [RFC6749]. 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

The following examples illustrate what a conforming Assertion and an access token request would look like.

The example shows an assertion issued and signed by the SAML Identity Provider identified as "https://saml-idp.example.com". The subject of the Assertion is identified by email address as "brian@example.com", who authenticated to the Identity Provider by means of a digital signature where the key was validated as part of an X.509 Public Key Infrastructure. The intended audience of the Assertion is "https://saml-sp.example.net", which is an identifier for a SAML Service Provider with which the authorization server identifies itself. The Assertion is sent as part of an access token request to the authorization server's token endpoint at "https://authz.example.net/token.oauth2".

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

```
<Assertion IssueInstant="2010-10-01T20:07:34.619Z"
  ID="ef1xsbZxPV2oqjd7HTLRLIB1Bb7"
  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>
    </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=PEFzc2VydGlvb3N1ZU1uc3RhbnQ9IjlmMTEtMDU
[...omitted for brevity...]aG5TdGF0ZW1lbnQ-PC9Bc3NlcnRpb24-
```

Figure 2: Example Request

5. Interoperability Considerations

Agreement between system entities regarding identifiers, keys, and endpoints is required in order to achieve interoperable deployments of this profile. Specific items that require agreement are as follows: values for the Issuer and Audience identifiers, the location of the token endpoint, the key used to apply and verify the digital signature over the Assertion, one-time use restrictions on Assertions, maximum Assertion lifetime allowed, and the specific Subject and attribute requirements of the Assertion. The exchange of such information is explicitly out of scope for this specification, and typical deployment of it will be done alongside existing SAML Web SSO deployments that have already established a means of exchanging such information. "Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0" [OASIS.saml-metadata-2.0-os] specifies one common method of exchanging SAML-related information about system entities.

The RSA-SHA256 algorithm, from [RFC6931], is a mandatory-to-implement XML signature algorithm for this profile.

6. Security Considerations

The security considerations described within the following specifications are all applicable to this document: "Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants" [RFC7521], "The OAuth 2.0 Authorization Framework" [RFC6749], and "Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0" [OASIS.saml-sec-consider-2.0-os].

The specification does not mandate replay protection for the SAML Assertion usage for either the authorization grant or for client authentication. It is an optional feature, which implementations may employ at their own discretion.

7. Privacy Considerations

A SAML Assertion may contain privacy-sensitive information and, to prevent disclosure of such information to unintended parties, should only be transmitted over encrypted channels, such as Transport Layer Security (TLS). In cases where it is desirable to prevent disclosure of certain information to the client, the Subject and/or individual attributes of a SAML Assertion should be encrypted to the authorization server.

Deployments should determine the minimum amount of information necessary to complete the exchange and include only that information in an Assertion (typically by limiting what information is included in an <AttributeStatement> or by omitting it altogether). In some cases, the Subject can be a value representing an anonymous or pseudonymous user, as described in Section 6.3.1 of the OAuth Assertion Framework [RFC7521].

8. IANA Considerations

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

This section registers the value "grant-type:saml2-bearer" in the IANA "OAuth URI" registry established by "An IETF URN Sub-Namespace for OAuth" [RFC6755].

- o URN: urn:ietf:params:oauth:grant-type:saml2-bearer
- o Common Name: SAML 2.0 Bearer Assertion Grant Type Profile for OAuth 2.0
- o Change Controller: IESG
- o Specification Document: RFC 7522

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

This section registers the value "client-assertion-type:saml2-bearer" in the IANA "OAuth URI" registry established by "An IETF URN Sub-Namespace for OAuth" [RFC6755].

- o URN: urn:ietf:params:oauth:client-assertion-type:saml2-bearer
- o Common Name: SAML 2.0 Bearer Assertion Profile for OAuth 2.0 Client Authentication
- o Change Controller: IESG
- o Specification Document: RFC 7522

9. References

9.1. Normative References

[OASIS.saml-core-2.0-os]

Cantor, S., Kemp, J., Philpott, R., and E. Maler, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-core-2.0-os, March 2005, <<http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>>.

[OASIS.saml-deleg-cs]

Cantor, S., Ed., "SAML V2.0 Condition for Delegation Restriction Version 1", Committee Specification 01, November 2009, <<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation-cs-01.html>>.

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

Hirsch, F., Philpott, R., and E. Maler, "Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-sec-consider-2.0-os, March 2005, <<http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<http://www.rfc-editor.org/info/rfc3986>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<http://www.rfc-editor.org/info/rfc4648>>.
- [RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <<http://www.rfc-editor.org/info/rfc6749>>.
- [RFC6931] Eastlake 3rd, D., "Additional XML Security Uniform Resource Identifiers (URIs)", RFC 6931, DOI 10.17487/RFC6931, April 2013, <<http://www.rfc-editor.org/info/rfc6931>>.
- [RFC7521] Campbell, B., Mortimore, C., Jones, M., and Y. Goland, "Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants", RFC 7521, DOI 10.17487/RFC7521, May 2015, <<http://www.rfc-editor.org/info/rfc7521>>.

9.2. Informative References

- [OASIS.WS-Fed]
Goodner, M. and A. Nadalin, "Web Services Federation Language (WS-Federation) Version 1.2", OASIS Standard, May 2009, <<http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html>>.
- [OASIS.WSS-SAMLTokenProfile]
Monzillo, R., Kaler, C., Nadalin, T., Hallam-Baker, P., and C. Milono, "Web Services Security SAML Token Profile Version 1.1.1", OASIS Standard, May 2012, <<http://docs.oasis-open.org/wss-m/wss/v1.1.1/wss-SAMLTokenProfile-v1.1.1.html>>.
- [OASIS.saml-metadata-2.0-os]
Cantor, S., Moreh, J., Philpott, R., and E. Maler, "Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-metadata-2.0-os, March 2005, <<http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf>>.

[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, <<http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>>.

[RFC6755] Campbell, B. and H. Tschofenig, "An IETF URN Sub-Namespace for OAuth", RFC 6755, DOI 10.17487/RFC6755, October 2012, <<http://www.rfc-editor.org/info/rfc6755>>.

[W3C.REC-html401-19991224]

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

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, Hannes Tschofenig, David Waite, Phil Hunt, and Mukesh Bhatnagar.

Authors' Addresses

Brian Campbell
Ping Identity

E-Mail: brian.d.campbell@gmail.com

Chuck Mortimore
Salesforce.com

E-Mail: cmortimore@salesforce.com

Michael B. Jones
Microsoft

E-Mail: mbj@microsoft.com

URI: <http://self-issued.info/>

부 록 1-1

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

지식재산권 협약서 정보

1-1.1 지식재산권 협약서(1) (스타일 적용-대항목/소항목)

- 해당 사항 없음

1-1.2 지식재산권 협약서(2) (스타일 적용-대항목/소항목)

- 해당 사항 없음

※ 상기 기재된 지식재산권 협약서 이외에도 본 표준이 발간된 후 접수된 협약서가 있을 수 있으니, TTA 웹사이트에서 확인하시기 바랍니다.

부 록 1-2

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

시험인증 관련 사항

1-2.1 시험인증 대상 여부 (스타일 적용-대항목/소항목)

- 해당 사항 없음

1-2.2 시험표준 제정 현황

- 해당 사항 없음

부 록 1-3

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

본 표준의 연계(family) 표준

- 해당 사항 없음

부 록 1-4

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

참고 문헌

- 해당 사항 없음

부 록 1-5

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

영문표준 해설서

SAML (Security Assertion Markup Language) 2.0은 신원 정보와 보안 정보를 여러 보안 도메인들간에 공유 할 수 있게 해주는 XML 기반의 프레임워크이다. SAML 표준은 주로 교차 도메인 웹 브라우저 SSO (Single Sign-On)를 제공하는 것을 목표로 하지만, 다른 컨텍스트에서도 쉽게 사용 할 수 있도록 모듈 식으로 확장 가능하게 설계되었다.

XML기반의 보안 토큰인 주장(Assertion)은 다른 프로토콜 및 표준에서 사용에 자주 채택 되는 SAML의 기본 구성물이다. 주장은 일반적으로 신원 공급자가 발급하고, 보안 목적으로 주장의 주체(subject)를 식별하기 위하여 그 내용에 의존하는 서비스 공급자에 의하여 사용(소비)된다.

공개인증 2.0의 인가 프레임워크[RFC6749]는 액세스 토큰을 사용하여 자원에 인증된 HTTP 요청을 작성하는 방법을 제공한다. 액세스 토큰은 자원 소유자의 승인을 받아 인가 서버에 의하여 제삼의 클라이언트에게 발급된다. 공개인증에서 인가(권한) 승인은 리소스 소유자 인가(권한)를 나타내는 중간 크리덴셜을 기술하는 데 사용되는 추상 용어이다. 인가 승인은 클라이언트가 액세스 토큰을 얻기 위해 사용한다. 여러 개의 인가 승인 유형은 광범위한 클라이언트 유형 및 사용자 경험을 지원하도록 정의되었다. 또한 공개인증은 추가 클라이언트를 지원하거나 다른 트러스트 프레임워크과의 교량을 제공하기 위하여 새로운 확장 승인 유형을 정의 하도록 고려 하였다. 마지막으로 공개인증이 인가 서버와 상호 작용할 때 클라이언트가 추가 인증 메커니즘을 정의 할 수 있도록 제공한다. 이 표준은 클라이언트가 인가 서버에서 직접적인 사용자 승인 단계없이, SAML 주장을 이용하여 신뢰 관계를 활용하고자 할 때 SAML 주장을 사용하여 액세스 토큰을 요청하는 방법을 정의한다. 또한 SAML 주장을 클라이언트 인증 메커니즘으로 사용하는 방법을 정의한다. 클라이언트 주장 인증은 클라이언트가 토큰 엔드포인트에 대해 인증 할 수 있는 대체 방법이기도 하고, 일부 인가 승인 유형과 함께 사용하여 완전하고 의미 있는 프로토콜 요청을 구성하기 위해 사용된다. 주장 인가 승인은 클라이언트 인증 또는 식별과 함께 사용되거나 또는 없이 사용될 수 있다. 지원되는 클라이언트 인증의 유형과 함께 클라이언트의 인증이 주장 인가 승인과 함께 필요한지에 대한 여부는 인가 서버의 재량에 따라 결정된다.

클라이언트가 인가 서버와 교환하거나 클라이언트 인증을 위해 SAML 주장을 사용하기

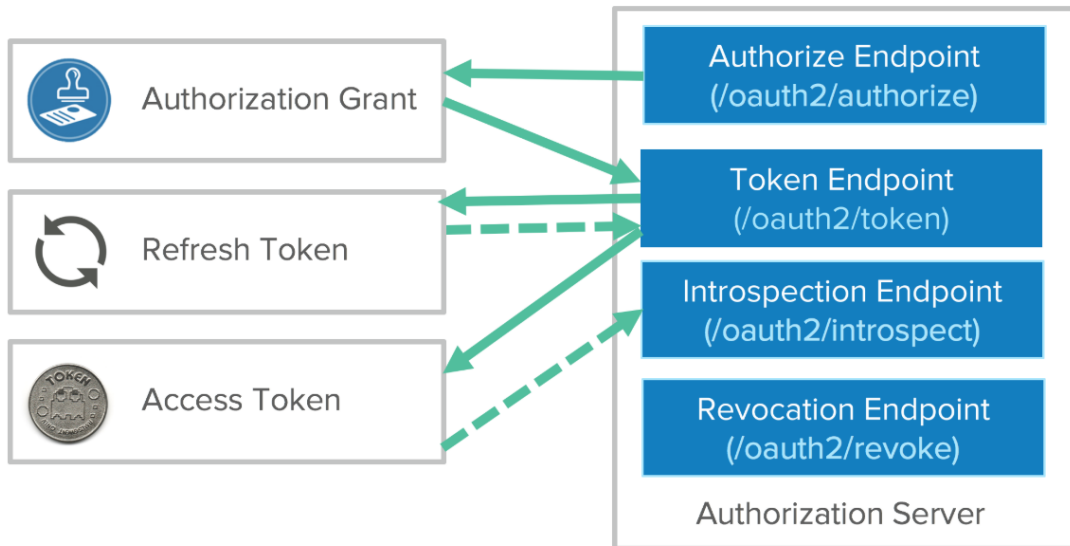
전에 클라이언트가 SAML 주장을 획득하는 프로세스는 표준의 범위가 아니다.

※ 주장(Assertion)은 여러 보안 도메인들에 걸쳐 신원 및 보안 정보를 쉽게 공유 할 수 있는 정보 상자(패키지)이다.

※ 인가 승인(Authorization grant)은 자원 소유자가 자원에 접근할 수 있는 권한을 부여 하였다는 확인증으로 클라이언트가 액세스 토큰을 요청하여 얻어오는데 사용한다. 인가 승인은 총 4개의 타입이 있다.

- Authorization Code 는 Client(클라이언트)가 Resource Owner(자원 소유자)에게 직접 권한 부여를 요청하는 대신, Resource Owner(자원 소유자)가 Authorization Server(권한 서버)에서 인증을 받고 권한을 허가 한다. 소유자가 권한을 허가하게 되면 Authorization Code(권한 코드)가 발급되고, 이 Authorization Code(권한 코드)를 클라이언트에게 전달하게 된다. 클라이언트는 이 코드를 권한 서버에 보내주면서 자신이 권한 허가를 받았다는 사실을 알리고 access token 을 받게 된다. 이 방법은 보안상 이점이 있다. access token 을 바로 Client(클라이언트)로 곧바로 전달하지 않기 때문에 전달과정에서 생길 수 있는 잠재적인 유출 위험을 방지하는데 도움을 준다.
- Implicit 는 Authorization Code(권한 코드)를 간소화한 절차이다. Authorization Code(권한 코드) 방식에서 access token 을 얻기 위한 중간 매개체로 Authorization Code(권한 코드)를 사용했던 것과 달리, 이 방식은 Authorization Code(권한 코드)가 별도로 발급되지 않고 access token 이 바로 발급된다. 과정이 줄어들어 편해지는데 대신 보안성은 낮아진다.
- Resource Owner Password Credentials 이 방식에서는 자원 소유자의 계정 아이디와 비밀번호 같은 계정 인증 정보가 access token 을 얻기 위한 Authorization Grant(권한 승인)로 사용된다. 계정정보를 애플리케이션에 직접 입력해야 하므로, 신뢰할 수 있어야 합니다. access token 을 얻은 후에는 리소스 요청을 위해 계정 아이디, 비밀번호를 Client(클라이언트)가 보관하고 있을 필요는 없다.
- Client Credentials 이 방식은 클라이언트 인증 방식 이라고도 합니다. 자원 소유자가 유저가 아닌, 클라이언트인 상황에서 활용되는 방식입니다. Client(클라이언트)가 관리하는 리소스에만 접근할 경우로 권한이 한정되어 있을 때 활용할 수 있다. 즉 Client(클라이언트)가 곧 Resource Owner(자원 소유자)가 되는 상황이다. Client(클라이언트)는 자기를 인증할 수 있는 정보를 Authorization Server(권한 서버)에 보내면서 access token 을 요청하게 된다.

※ RFC6749에서 인가 프로세스는 두 개의 인가 서버 엔드포인트 (HTTP 자원)를 사용한다. 인증 엔드포인트는 클라이언트가 사용자 에이전트 재 지정을 통해 자원 소유자로부터 권한을 얻기 위해 사용하고, 토큰 엔드포인트는 클라이언트가 일반적으로 클라이언트 인증을 사용하여 액세스 토큰에 대한 인가 승인을 교환하는 데 사용된다. 이외에도 확장 유형에 따라 추가적 엔드포인트를 정의 할 수 있다.



(그림I-5.1) 인가(Authorization) 서버의 엔드포인트 구성 예

부 록 1-6

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

표준의 이력

판수	채택일	표준번호	내용	담당 위원회
제1판			공개인증2.0 클라이언트 인증 및 인가 승인을 위한 SAML2.0 프로파일	PG504