

OAuth Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 22, 2015

M. Jones
A. Nadalin
C. Baker
Microsoft
August 21, 2014

OAuth 2.0 Token Exchange
draft-ietf-oauth-token-exchange-00

Abstract

This specification defines how to request and obtain Security Tokens from OAuth Authorization Servers, including enabling one party to act on behalf of another or enabling one party to delegate authority to another.

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 February 22, 2015.

Copyright Notice

Copyright (c) 2014 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](#) [3](#)
- [1.1. Requirements Notation and Conventions](#) [3](#)
- [1.2. Terminology](#) [3](#)
- [1.3. On-Behalf-Of vs. Impersonation Semantics](#) [3](#)
- [2. Security Token Request](#) [4](#)
- [2.1. Act-As Security Token Requests](#) [5](#)
- [2.2. On-Behalf-Of Security Token Requests](#) [6](#)
- [3. Security Token Response](#) [6](#)
- [4. Conveying Eligibility to Act As Another Party](#) [7](#)
- [5. Implementation Considerations](#) [8](#)
- [6. IANA Considerations](#) [8](#)
- [7. Security Considerations](#) [8](#)
- [8. References](#) [8](#)
- [8.1. Normative References](#) [8](#)
- [8.2. Informative References](#) [8](#)
- [Appendix A. Open Issues](#) [9](#)
- [Appendix B. Acknowledgements](#) [9](#)
- [Appendix C. Document History](#) [9](#)
- [Authors' Addresses](#) [10](#)

1. Introduction

This specification defines how to request and obtain Security Tokens from OAuth Authorization Servers [[RFC6749](#)], including enabling one party to act on behalf of another or enabling one party to delegate authority to another. The functionality defined and the terminology used in this specification are intentionally parallel to the functionality and terminology defined by [[WS-Trust](#)], including On-Behalf-Of and Act-As.

A Security Token is a set of information that facilitates the sharing of identity and security information across security domains. Examples of Security Tokens include JSON Web Tokens (JWTs) [[JWT](#)] and SAML Assertions [[OASIS.saml-core-2.0-os](#)]. Security Tokens are typically signed to achieve integrity and sometimes also encrypted to achieve confidentiality. Security Tokens are also described as Assertions in [[I-D.ietf-oauth-assertions](#)].

This specification defines a new Security Token Request Grant Type used at the Token Endpoint to convey the parameters for a Security Token request and Security Token response parameter used in responses to these requests. The Security Token Request is a JSON Web Token (JWT) [[JWT](#)] that is issued by the requesting party that contains parameters of the request as Claims.

The Security Tokens obtained could be used in a number of contexts, the specifics of which are beyond the scope of this specification. Examples include using them with the

1.1. Requirements Notation and 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](#)].

1.2. Terminology

This specification uses the terms "Authorization Server" "Token Endpoint", "Token Request", and "Token Response" defined by OAuth 2.0 [[RFC6749](#)], and the terms "Claim" and "JWT Claims Set" defined by JSON Web Token (JWT) [[JWT](#)].

1.3. On-Behalf-Of vs. Impersonation Semantics

When principal A acts on behalf of principal B, A is given all the rights that B has within some defined rights context. Whereas, with on-behalf-of semantics, principal A still has its own identity separate from B and it is explicitly understood that while B may have

delegated its rights to A, any actions taken are being taken by A and not B. In a sense, A is an agent for B.

On-behalf-of semantics are therefore different than impersonation semantics, with which it is sometimes confused. When principal A impersonates principal B, then in so far as any entity receiving Claims is concerned, they are actually dealing with B. It is true that some members of the identity system might have awareness that impersonation is going on but it is not a requirement. For all intents and purposes, when A is acting for B, A is B.

2. Security Token Request

A Security Token Request is sent to the Token Endpoint as a Token Request message using this Grant Type value:

```
urn:ietf:params:oauth:grant-type:security-token-request
  Grant Type value indicating that this Token Request is a Security
  Token Request.
```

A Token Request parameter with a related name is used to convey the information contained in Security Token Request as a JWT:

```
security_token_request
  Token Request parameter whose value is a JWT containing the
  Security Token Request information.
```

An example Security Token Request (with extra line breaks for display purposes only) follows:

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

grant_type=urn:ietf:params:oauth:grant-type:security-token-request
&security_token_request=eyJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJ...
[omitted for brevity]
```

The "security_token_request" parameter value is a JWT with the following members:

```
iss
  REQUIRED. Issuer of the principal requesting the Security Token.
```


sub

REQUIRED. Identifier of the principal requesting the Security Token at the issuer.

security_token_type

OPTIONAL. Identifier for the type of the requested Security Token. If not present, the default is that a JWT is being requested. A JWT can also be requested with the identifier "urn:ietf:params:oauth:token-type:jwt".

scopes

OPTIONAL. Array of strings, each of which represents a service context that the requested Security Token is being requested to be used for. The array MUST contain at least one scope value. The definition of these contexts is outside the scope of this specification. (Note: This request element serves the same purpose as the WS-Trust AppliesTo RST element.)

The request JWT MUST be signed by the issuer so the identity of the requesting party can be validated unless the identity of the requesting party is known to the Authorization Server by other means; in that case, the JWT can use the "alg" value "none".

The following is an example of a JWT Claims Set for a Security Token Request:

```
{
  "iss": "https://server.example.com",
  "sub": "24400320",
  "scopes": ["example"]
}
```

2.1. Act-As Security Token Requests

This specification defines the ability to request a Security Token for the requesting party to use to act as the specified party. This is accomplished using this Token Request parameter:

act_as

This OPTIONAL request parameter indicates that the requested Security Token is expected to contain information about the identity represented by the Security Token that is the value of this parameter, enabling the requesting party to use the returned Security Token to act as this identity.

The following is an example of a JWT Claims Set for a Security Token Request using an "act_as" Claim:

```
{
  "iss": "https://server.example.com",
  "sub": "24400320",
  "scopes": ["example"],
  "act_as": "eyJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJ ..."
}
```

2.2. On-Behalf-Of Security Token Requests

This specification defines the ability to request a Security Token on behalf of another party. This is accomplished using this Token Request parameter:

on_behalf_of

This OPTIONAL request parameter indicates that the Security Token is being requested on behalf of another party. The identity of the party upon whose behalf the request is being made is represented by the Security Token that is the value of this parameter. Proof of eligibility to act on behalf of that identity MAY be conveyed by including an "actor" Claim identifying the requesting party in the Security Token, per [Section 4](#), provided the Security Token is a JWT.

The following is an example of a JWT Claims Set for a Security Token Request using an "on_behalf_of" Claim:

```
{
  "iss": "https://server.example.com",
  "sub": "24400320",
  "scopes": ["example"],
  "on_behalf_of": "eyJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJ ..."
}
```

3. Security Token Response

A Security Token Response is returned from the Token Endpoint as a Token Response message containing these members:

security_token

Returned Security Token.

security_token_type

Identifier for the type of the returned Security Token. If the Security Token is a JWT, this identifier is "urn:ietf:params:oauth:token-type:jwt".

An example successful response is as follows:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "security_token": "eyJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJ ...",
  "security_token_type": "urn:ietf:params:oauth:token-type:jwt"
}
```

4. Conveying Eligibility to Act As Another Party

It is useful to be able to make a statement that one party is authorized to act on behalf of another party. This can be done by having the party being acted for issue a Security Token containing a Claim identifying the party that will act for it as an authorized actor. This statement can also optionally identify scopes in which the actor is eligible to act through another Claim. The following Claims are defined for use in JWTs for these purposes:

actor

Security Token that identifies a party who is asserted as being eligible to act for the party identified by the JWT containing this Claim.

scopes

OPTIONAL. Array of strings, each of which represents a service context for which the actor is asserted as being eligible to act for the party identified by the JWT containing this Claim. The array MUST contain at least one scope value. The definition of these contexts is outside the scope of this specification.

The JWT issued by the party being acted for MUST be signed so the identity of the party being acted for can be validated unless the identity of the party being acted for is known to the Authorization Server by other means; in that case, the JWT can use the "alg" value "none".

5. Implementation Considerations

Implementations of the specification MUST implement support for using JWTs as the Security Tokens. Other Security Token types MAY be supported.

6. IANA Considerations

The "urn:ietf:params:oauth:grant-type:security-token-request" Grant Type is to be registered in the IANA urn:ietf:params:oauth registry established in An IETF URN Sub-Namespace for OAuth [[RFC6755](#)].

The "scopes", "act_as", and "on_behalf_of" Claims are to be registered in the JSON Web Token Claims registry.

7. Security Considerations

All of the normal security issues, especially in relationship to comparing URIs and dealing with unrecognized values, that are discussed in JWT [[JWT](#)] also apply here.

In addition, on-behalf-of introduces its own unique security issues. Any time one principal is delegated the rights of another principal, the potential for abuse is always a concern. That is why use of the "scopes" member is suggested. The scope values restrict the contexts in which the delegated rights can be exercised.

8. References

8.1. Normative References

- [JWT] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", [draft-ietf-oauth-json-web-token](#) (work in progress), July 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework", [RFC 6749](#), October 2012.

8.2. Informative References

- [I-D.ietf-oauth-assertions]
Campbell, B., Mortimore, C., Jones, M., and Y. Goland,

"Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants", [draft-ietf-oauth-assertions](#) (work in progress), April 2014.

[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.

[RFC6755] Campbell, B. and H. Tschofenig, "An IETF URN Sub-Namespace for OAuth", [RFC 6755](#), October 2012.

[WS-Trust]

Nadalin, A., Goodner, M., Gudgin, M., Barbir, A., and H. Granqvist, "WS-Trust 1.4", February 2012, <<http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.html>>.

[Appendix A](#). Open Issues

The following decisions need to be made and updates on this spec performed:

- o Should we say anything about proof of possession of the target party's key in the On-Behalf-Of case beyond specifying the use of the "actor" Claim?
- o Revise the text in the On-Behalf-Of vs. Impersonation Semantics section to better align the terminology used with the semantics specified.
- o Address the sources of potential terminological confusion discussed in John Bradley's review comments.
- o Add examples illustrating concrete uses of act-as and on-behalf-of.

[Appendix B](#). Acknowledgements

The authors wish to thank Brian Campbell and John Bradley for reviews of the specification.

[Appendix C](#). Document History

[[to be removed by the RFC Editor before publication as an RFC]]

-00

- o Created initial working group draft from [draft-jones-oauth-token-exchange-01](#).

Authors' Addresses

Michael B. Jones
Microsoft

Email: mbj@microsoft.com

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

Anthony Nadalin
Microsoft

Email: tonynad@microsoft.com

Caleb Baker
Microsoft

Email: calebb@microsoft.com

