Network Working Group	M.B. Jones
Internet-Draft	Microsoft
Intended status: Standards Track	March 28, 2011
Expires: September 29, 2011	

JSON Web Token (JWT) Bearer Profile for OAuth 2.0
draft-jones-oauth-jwt-bearer-00

Abstract

This specification defines the use of a JSON Web Token (JWT) bearer token as a means of requesting an OAuth 2.0 access token.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

JSON Web Token (JWT) [JWT] is a JSON-based security token encoding that enables identity and security information to be shared across security domains. JWTs utilize JSON data structures, as defined in RFC 4627 *[RFC4627]*.

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.

This specification defines an extension grant type that profiles the use of a JSON Web Token (JWT) in requesting an OAuth 2.0 access token.

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 [RFC2119].
Unless otherwise noted, all the protocol parameter names and values are case sensitive.

1.2. Terminology

All terms are as defined in [I-D.ietf-oauth-v2] and [JWT].

2. JWT Access Token Request

A JSON Web Token (JWT) bearer token can be used to request an access token when a client wishes to utilize an existing trust relationship, expressed through the semantics of the JWT, without a direct user approval step at the authorization server.
The process by which the client obtains the JWT, prior to exchanging it with the authorization server, is out of scope.

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

*(A) The client sends an access token request to the authorization server that includes a JWT bearer token and a grant_type of http://oauth.net/grant_type/jwt/1.0/bearer.

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

2.1. Client Requests Access Token

The client includes the JWT in the access token request, the core details of which are defined in OAuth [I-D.ietf-oauth-v2], by specifying http://oauth.net/grant_type/jwt/1.0/bearer as the absolute URI value of the grant_type parameter and by adding the following parameter:

jwt

REQUIRED. The value of the jwt parameter MUST be a single JWT that is represented using the Compact Serialization.

scope

OPTIONAL. The scope of the access request expressed as a list of space-delimited strings. The value is defined by the authorization server. If the value contains multiple space-delimited strings, their order does not matter, and each string adds an additional access range to the requested scope.

2.2. JWT Content and Processing Requirements

Prior to issuing an access token response as described in [I-D.ietf-oauth-v2], the authorization server MUST validate the JWT according to the criteria below. If present, the authorization server MUST also validate the client credentials. Application of additional restrictions and policy are at the discretion of the authorization server.

- The JWT MUST contain an iss (issuer) claim that contains a unique identifier for the entity that issued the JWT.
- The JWT MUST contain a prn (principal) claim. The principal MUST identify the resource owner for whom the access token is being requested.
- The JWT MUST contain an aud (audience) claim containing a URI reference that identifies the authorization server as the intended audience. The authorization server MUST verify that it is an intended audience for the JWT.
- The JWT MUST contain an exp (expiration) claim that limits the time window during which the JWT can be used. The authorization server MUST verify that the expiration time has not passed, subject to allowable clock skew between systems. The authorization server MAY reject JWTs with an exp claim value that is unreasonably far in the future.
- The JWT MAY contain an iat (issued at) claim containing the UTC time at which the JWT was issued. This time is represented as an IntDate, as defined by [JWT].
- The JWT MAY contain other claims.
- The JWT MUST be digitally signed by the issuer in the manner described in the JWT specification and the authorization server MUST verify the signature.
- The authorization server MUST verify that the JWT is valid in all other respects per [JWT].

2.3. Error Response

If the JWT is not valid or has expired, 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 JWT 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 JWT and access token request would look like.
Below is an example JSON object that could be encoded to produce the JWT Claims Object for a JWT:

```
{"iss":"https://jwt-idp.example.com",
 "prn":"mailto:mike@example.com",
 "aud":"https://jwt-rp.example.net",
 "iat":1300815780,
 "exp":1300819380,
 "http://claims.example.com/member":true}
```

The following example JSON object, used as the header of a JWT, declares that the JWT is signed with the ECDSA P-256 SHA-256 algorithm.

```
{"alg":"ES256"}
```

To present a JWT with the claims and header shown above 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%2Fjwt%2F1.0%2Fbearer&
jwt=eyJhbGciOiJFUzI1NiJ9.eyJpc3Mi[...omitted for brevity...].
J9l-ZhwP_2n[...omitted for brevity...]
```

3. Security Considerations

Authorization servers SHOULD issue access tokens with a limited lifetime and require clients to refresh them by requesting a new access token using the same JWT, if it is still valid, or with a new JWT. The authorization server SHOULD NOT issue a refresh token.

4. IANA Considerations

4.1. OAuth Parameters Registration

This specification registers the following parameters in the OAuth Parameters Registry established by [I-D.ietf-oauth-v2].

4.1.1. The "jwt" OAuth Parameter

Parameter name: jwt

Parameter usage location: token request

Change controller: IETF

Specification document(s): [[this document]]

Related information: None

5. References

5.1. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC4627]	Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.
[I-D.ietf-oauth-v2]	Hammer-Lahav, E, Recordon, D and D Hardt, "The OAuth 2.0 Authorization Protocol", Internet-Draft draft-ietf-oauth-v2-13, February 2011.
[JWT]	Jones, M.B., Balfanz, D., Bradley, J., Goland, Y.Y., Panzer, J., Sakimura, N. and P. Tarjan, "JSON Web Token (JWT)", March 2011.

5.2. Informative References

[I-D.ietf-oauth-saml2-bearer]	Campbell, B and C Mortimore, "SAML 2.0 Bearer Assertion Grant Type Profile for OAuth 2.0", Internet-Draft draft-ietf-oauth-saml2-bearer-03, February 2011.

Appendix A. Acknowledgements

This profile was derived from the SAML2 Bearer Assertion Grant Type Profile for OAuth 2.0 [I-D.ietf-oauth-saml2-bearer] by Brian Campbell and Chuck Mortimore.

Appendix B. Document History

[[to be removed by RFC editor before publication as an RFC]]
-00

*Initial draft.

Author's Address

Michael B. Jones Jones Microsoft EMail: mbj@microsoft.com URI: http://self-issued.info/