JWT Response for OAuth Token Introspection

draft-lodderstedt-oauth-jwt-introspection-response-01

Abstract

This draft proposes an additional JSON Web Token (JWT) based response for OAuth 2.0 Token Introspection.

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 https://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 November 29, 2018.

Copyright Notice

Copyright (c) 2018 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 (https://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.
1. Introduction

OAuth 2.0 Token Introspection [RFC7662] specifies a method for a protected resource to query an OAuth 2.0 authorization server to determine the state of an access token and obtain data associated with the access token. This allows deployments to implement identifier-based access tokens in an interoperable way.

The introspection response as specified in OAuth 2.0 Token Introspection [RFC7662] is a plain JSON object. However, there are use cases where the resource server requires stronger assurance that the authorisation server issued the access token, including cases where the authorisation server assumes liability for the token’s content. An example is a resource server using verified person data to create qualified electronic signatures.

In such use cases, it would be useful to return a signed JWT as the introspection response. This specification extends the Token Introspection endpoint with the capability to return responses as JWTs.

2. Requesting a JWT Response

A resource server requests to receive a JWT introspection response by including an Accept header with content type "application/jwt" in the introspection request.
The following is a non-normative example request:

```plaintext
POST /introspect HTTP/1.1
Host: server.example.com
Accept: application/jwt
Content-Type: application/x-www-form-urlencoded

token=2YotnFZFEjr1zCsicMWpAA
```

3. JWT Response

The introspection endpoint responds with a JWT, setting the Content-Type header to "application/jwt".

This JWT MUST contain the claims "iss" and "aud" in order to prevent misuse of the JWT as ID or access token (see Section 8.1).

This JWT may furthermore contain all other claims described in Section 2.2. of [RFC7662].

The following is a non-normative example response (with line breaks for display purposes only):

```plaintext
HTTP/1.1 200 OK
Content-Type: application/jwt

eyJraWQiOiIxIiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiJaNU8zdXBQQzg4UXJBAngwMGRpcyIsImFlZCI6Imh0dHBzOlwvXC9wcm9Qnot0ZWN0ZWQuZXhhbXBsb2d1XCI6XCI6Li4iLCJhY3RpdmUiOnRydWUsImV4cCI6MTQxOTM1NjIzOCwiZXhwIjoxNDE5MzUwMjM4LCJhbGciOiJ0c2VfX21hY2tfb2x1bWIiLCJpZCI6MTQxOTM1NzYyOiwicGF5ZCI6Q0EiLCJpYXQiOjE2NDQwMjM4NjIiLCJwdXIiOiJ1c2VyIiwicGF5ZHMiOnRydWUsImlhdCI6MTQxOTM1NjIzOCwiYWN0aW9uIjoiTWF0aXZpY2FsYSIsImV4cCI6NiwiYnVvdGQiOiI4OWZhODE2OC03MDUtZGJhNy04ZjEtNjI0YWQ1NzRjODAzOCJ9
```

The example response contains the following JSON document:
4. Client Metadata

The authorization server determines what algorithm to employ to secure the JWT for a particular introspection response. This decision can be based on registered metadata parameters for the resource server, supplied via dynamic client registration with the resource server posing as the client.

The parameter names follow the pattern established by OpenID Connect Dynamic Client Registration [OpenID.Registration] for configuring signing and encryption algorithms for JWT responses at the UserInfo endpoint.

The following client metadata parameters are introduced by this specification:

introspection_signed_response_alg  JWS [RFC7515] "alg" algorithm JWA [RFC7518] REQUIRED for signing introspection responses. If this is specified, the response will be JWT [RFC7519] serialized, and signed using JWS. The default, if omitted, is for the introspection response to return the Claims as a UTF-8 encoded JSON object using the "application/json" content type, as defined in [RFC7662].

introspection_encrypted_response_alg  JWE [RFC7516] "alg" algorithm JWA [RFC7518] REQUIRED for encrypting introspection responses. If both signing and encryption are requested, the response will be signed then encrypted, with the result being a Nested JWT, as defined in JWT [RFC7519]. The default, if omitted, is that no encryption is performed.

introspection_encrypted_response_enc  JWE [RFC7516] "enc" algorithm JWA [RFC7518] REQUIRED for encrypting introspection responses. If "introspection_encrypted_response_alg" is
specified, the default for this value is A128CBC-HS256. When "introspection_encrypted_response_enc" is included, "introspection_encrypted_response_alg" MUST also be provided.

Resource servers may register their public encryption keys using the "jwks_uri" or "jwks" metadata parameters.

5. Authorization Server Metadata

Authorization servers SHOULD publish the supported algorithms for signing and encrypting the JWT of an introspection response by utilizing OAuth Authorization Server Metadata parameters.

The following parameters are introduced by this specification:

introspection_signing_alg_values_supported  OPTIONAL. JSON array containing a list of the JWS [RFC7515] signing algorithms ("alg" values) JWA [RFC7518] supported by the Introspection Endpoint to sign the response.

introspection_encryption_alg_values_supported  OPTIONAL. JSON array containing a list of the JWE [RFC7516] encryption algorithms ("alg" values) JWA [RFC7518] supported by the Introspection Endpoint to encrypt the response.

introspection_encryption_enc_values_supported  OPTIONAL. JSON array containing a list of the JWE [RFC7516] encryption algorithms ("enc" values) JWA [RFC7518] supported by the Introspection Endpoint to encrypt the response.

6. Acknowledgements

We would like to thank Petteri Stenius and Neil Madden for their valuable feedback.

7. IANA Considerations

7.1. OAuth Dynamic Client Registration Metadata Registration

This specification requests registration of the following client metadata definitions in the IANA "OAuth Dynamic Client Registration Metadata" registry [IANA.OAuth.Parameters] established by [RFC7591]:

7.1.1. Registry Contents

- Client Metadata Name: "introspection_signed_response_alg"
o Client Metadata Description: String value indicating the client’s desired introspection response signing algorithm.

o Change Controller: IESG

o Specification Document(s): Section 4 of [[ this specification ]]

o Client Metadata Name: "introspection_encrypted_response_alg"

o Client Metadata Description: String value specifying the desired introspection response encryption algorithm (alg value).

o Change Controller: IESG

o Specification Document(s): Section 4 of [[ this specification ]]

o Client Metadata Name: "introspection_encrypted_response_enc"

o Client Metadata Description: String value specifying the desired introspection response encryption algorithm (enc value).

o Change Controller: IESG

o Specification Document(s): Section 4 of [[ this specification ]]

7.2. OAuth Authorization Server Metadata Registration

This specification requests registration of the following value in the IANA "OAuth Authorization Server Metadata" registry [IANA.OAuth.Parameters] established by [I-D.ietf-oauth-discovery].

7.2.1. Registry Contents

o Metadata Name: "introspectionSigningAlgsSupported"

o Metadata Description: JSON array containing a list of algorithms supported by the authorization server for introspection response signing.

o Change Controller: IESG

o Specification Document(s): Section 5 of [[ this specification ]]

o Metadata Name: "introspectionEncryptionAlgsSupported"

o Metadata Description: JSON array containing a list of algorithms supported by the authorization server for introspection response encryption (alg value).
7.3. OAuth Token Introspection Response

TBD: add all OpenID Connect standard claims.

8. Security Considerations

8.1. Cross-JWT Confusion

JWT introspection responses and OpenID Connect ID Tokens are syntactically more or less equivalent. An attacker could therefore try to misuse an JWT obtained from an introspection response to impersonate the user whose claims are included in this JWT at a OpenID Connect RP. Such an attack is treated and prevented like any other token substitution attack. The AS MUST include the claims "iss" and "aud" into every JWT introspection response. This allows every well behaving OpenID Connect RP to detect substitution by checking the "iss" and "aud" claims as described in Section 3.1.3.7. of [OpenID.Core]. RPs should also use and check the "nonce" parameter and claim to prevent token and code replay.

Resource servers utilizing JWTs to represent structured access tokens could be susceptible to replay attacks as well. Resource servers should therefore apply proper counter measures against replay as described in [I-D.ietf-oauth-security-topics], section 2.2.

JWT Confusion and other attacks on JJWTs are discussed in detail in [I-D.ietf-oauth-jwt-bcp].

9. References
9.1. Normative References

[I-D.ietf-oauth-discovery]  

[I-D.ietf-oauth-jwt-bcp]  

[I-D.ietf-oauth-security-topics]  

[OpenID.Core]  
NRI, Ping Identity, Microsoft, Google, and Salesforce, "OpenID Connect Core 1.0 incorporating errata set 1", Nov 2014, <http://openid.net/specs/openid-connect-core-1_0.html>.

[OpenID.Registration]  
NRI, Ping Identity, and Microsoft, "OpenID Connect Dynamic Client Registration 1.0 incorporating errata set 1", Nov 2014, <https://openid.net/specs/openid-connect-registration-1_0.html>.

[RFC2119]  

[RFC2246]  

[RFC7515]  

[RFC7516]  
9.2. Informative References

[IANA.OAuth.Parameters]  
IANA, "OAuth Parameters",  
<http://www.iana.org/assignments/oauth-parameters>.

Appendix A. Document History

-01

- fixed typos in client meta data field names
- added OAuth Server Metadata parameters to publish algorithms supported for signing and encrypting the introspection response
- added registration of new parameters for OAuth Server Metadata and Client Registration
- added explicit request for JWT introspection response
- made iss and aud claims mandatory in introspection response
- Stylistic and clarifying edits, updates references

-00

- initial version
Authors’ Addresses

Torsten Lodderstedt (editor)
YES.com AG

Email: torsten@lodderstedt.net

Vladimir Dzhuvinov
Connect2id Ltd.

Email: vladimir@connect2id.com