Open Authentication Protocol
Internet-Draft
Intended status: Standards Track
Expires: November 29, 2018

T. Lodderstedt, Ed.
YES.com AG
V. Dzhuvinov
Connect2id Ltd.
May 28, 2018

# 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.

Table of Contents



## 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:

```
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):

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

eyJraWQiOiIxIiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiJaNU8zdXBQQzg4UXJBa
ngwMGRpcyIsImF1ZCI6Imh0dHBzOlwvXC9wcm90ZWN0ZWQuZXhhbXBsZS5uZXRcL
3Jlc291cmNlIiwiZXh0ZW5zaW9uX2ZpZWxkIjoidHdlbnR5LXNldmVuIiwic2Nvc
GUiOiJyZWFkIHdyaXRlIGRvbHBoaW4iLCJpc3MiOiJodHRwczpcL1wvc2VydmVyL
mV4YW1wbGUuY29tXC8iLCJhY3RpdmUiOnRydWUsImV4cCI6MTQxOTM1NjIzOCwia
WF0IjoxNDE5MzUwMjM4LCJjbGllbnRfaWQiOiJsMjM4ajMyM2RzLTIzaWo0Iiwid
XNlcm5hbWUiOiJqZG9lIn0.HEQHf05vqVvWVnWuEjbzUnPz6JDQVR69QkxgzBNq5
kk-sK54ieg1STazXGsdFAT8nUhiiV1f_Z4HOKNnBs8TLKaFXokhA0MqNBOYI--2u
nVHDqI_RPmC3p0NmP02Xmv4hzxFmTmpgjSy3vpKQDihOjhwNBh7G81JNaJqjJQTR
v_1dHUPJotQjMK3k8_5FyiO2p64Y2VyxyQn1VWVlgOHlJwhj6BaGHk4Qf5F8DHQZ
1WCPg2p_-hwfINfXh1_buSjxyDRF4oe9pKy6ZB3ejh9qIMm-WrwltuU1uWMXxN6e
S6tUtpKo8UCHBwLWCHmJN7KU6ZojmaISspdS23lELAlyw
```

The example response contains the following JSON document:

```
{
  "sub": "Z5O3upPC88QrAjx00dis",
  "aud": "https://protected.example.net/resource",
  "scope": "read write dolphin",
  "iss": "https://server.example.com/",
  "active": true,
  "exp": 1419356238,
  "iat": 1419350238,
  "client_id": "l238j323ds-23ij4",
  "given_name": "John",
  "family_name":"Doe",
  "birthdate":"1982-02-01"
}
```

## 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"

- Client Metadata Description: String value indicating the client's desired introspection response signing algorithm.
- Change Controller: IESG
- Specification Document(s): Section 4 of [[ this specification ]]
- Client Metadata Name: "introspection_encrypted_response_alg"
- Client Metadata Description: String value specifying the desired introspection response encryption algorithm (alg value).
- Change Controller: IESG
- Specification Document(s): Section 4 of [[ this specification ]]
- Client Metadata Name: "introspection_encrypted_response_enc"
- Client Metadata Description: String value specifying the desired introspection response encryption algorithm (enc value).
- Change Controller: IESG
- 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

- Metadata Name: "introspection_signing_alg_values_supported"
- Metadata Description: JSON array containing a list of algorithms supported by the authorization server for introspection response signing.
- Change Controller: IESG
- Specification Document(s): Section 5 of [[ this specification ]]
- Metadata Name: "introspection_encryption_alg_values_supported"
- Metadata Description: JSON array containing a list of algorithms supported by the authorization server for introspection response encryption (alg value).

- Change Controller: IESG
- Specification Document(s): Section 5 of [[ this specification ]]
- Metadata Name: "introspection_encryption_enc_values_supported"
- Metadata Description: JSON array containing a list of algorithms supported by the authorization server for introspection response encryption (enc value).
- Change Controller: IESG
- Specification Document(s): Section 5 of [[ this specification ]]

### 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 JWTs are discussed in detail in [I-D.ietf-oauth-jwt-bcp].

## 9. References

### 9.1. Normative References

[I-D.ietf-oauth-discovery]
Jones, M., Sakimura, N., and J. Bradley, "OAuth 2.0 Authorization Server Metadata", draft-ietf-oauth-discovery-10 (work in progress), March 2018.

[I-D.ietf-oauth-jwt-bcp]
Sheffer, Y., Hardt, D., and M. Jones, "JSON Web Token Best Current Practices", draft-ietf-oauth-jwt-bcp-03 (work in progress), May 2018.

[I-D.ietf-oauth-security-topics]
Lodderstedt, T., Bradley, J., Labunets, A., and D. Fett, "OAuth 2.0 Security Best Current Practice", draft-ietf-oauth-security-topics-06 (work in progress), May 2018.

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

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, DOI 10.17487/RFC2246, January 1999, <https://www.rfc-editor.org/info/rfc2246>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <https://www.rfc-editor.org/info/rfc7516>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <https://www.rfc-editor.org/info/rfc7518>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC7591] Richer, J., Ed., Jones, M., Bradley, J., Machulak, M., and P. Hunt, "OAuth 2.0 Dynamic Client Registration Protocol", RFC 7591, DOI 10.17487/RFC7591, July 2015, <https://www.rfc-editor.org/info/rfc7591>.

[RFC7662] Richer, J., Ed., "OAuth 2.0 Token Introspection", RFC 7662, DOI 10.17487/RFC7662, October 2015, <https://www.rfc-editor.org/info/rfc7662>.

### 9.2. Informative References

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

## Appendix A. Document History

[[ To be removed from the final specification ]]

-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