Open Authentication Protocol
Internet-Draft
Intended status: Standards Track
Expires: October 27, 2020

T. Lodderstedt, Ed.
yes.com AG
V. Dzhuvinov
Connect2id Ltd.

April 25, 2020

JWT Response for OAuth Token Introspection
draft-ietf-oauth-jwt-introspection-response-09

Abstract

This specification proposes an additional JSON Web Token (JWT) secured 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 October 27, 2020.

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 enables deployments to implement opaque 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 authorization server issued the token introspection response for an access token, including cases where the authorization server assumes liability for the content of the token introspection response. An example is a resource server using verified person data to create certificates, which in turn are used to create qualified electronic signatures.

In such use cases it may be useful or even required to return a signed JWT [RFC7519] as the introspection response. This specification extends the token introspection endpoint with the capability to return responses as JWTs.

2. Requirements Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Resource Server Management

The authorization server (AS) and the resource server (RS) maintain a strong two-way trust relationship. The resource server relies on the authorization server to obtain authorization, user and other data as input to its access control decisions and service delivery. The authorization server relies on the resource server to handle the provided data appropriately.

In the context of this specification, the Token Introspection Endpoint is used to convey such security data and potentially also privacy sensitive data related to an access token.

In order to process the introspection requests in a secure and privacy-preserving manner, the authorization server MUST be able to identify, authenticate and authorize resource servers.

To support encrypted token introspection response JWTs, the authorization server MUST also be provided with the respective resource server encryption keys and algorithms.

The authorization server MUST be able to determine whether an RS is the audience for a particular access token and what data it is entitled to receive, otherwise the RS is not authorized to obtain data for the access token. The AS has the discretion how to fulfil this requirement. The AS could, for example, maintain a mapping between scopes values and resource servers.

The requirements given above imply that the authorization server maintains credentials and other configuration data for each RS.

One way is by utilizing dynamic client registration [RFC7591] and treating every RS as an OAuth client. In this case, the authorization server is assumed to at least maintain "client_id" and "token_endpoint_auth_method" with complementary authentication method

metadata, such as "jwks" or "client_secret". In cases where the AS needs to acquire consent to transmit data to a RS, the following client metadata fields are recommended: "client_name", "client_uri", "contacts", "tos_uri", "policy_uri".

The AS MUST restrict the use of client credentials by a RS to the calls it requires, e.g. the AS MAY restrict such a client to call the token introspection endpoint only. How the AS implements this restriction is beyond the scope of this specification.

This specification further introduces client metadata to manage the configuration options required to sign and encrypt token introspection response JWTs.

4. Requesting a JWT Response

A resource server requests a JWT introspection response by including an "Accept" HTTP header "application/token-introspection+jwt" in the introspection request.

The AS SHOULD authenticate the caller at the token introspection endpoint. Authentication can utilize client authentication methods or a separate access token issued to the resource server. Whether a resource server is required to authenticate is determined by the respective RS-specific policy at the AS.

The following is a non-normative example request with client authentication:

```
POST /introspect HTTP/1.1
Host: as.example.com
Accept: application/token-introspection+jwt
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded

token=2YotnFZFEjr1zCsicMWpAA
```

5. JWT Response

The introspection endpoint responds with a JWT, setting the "Content-Type" HTTP header to "application/token-introspection+jwt" and the JWT "typ" ("type") header to "token-introspection+jwt".

The JWT MUST include the following top-level claims:

iss MUST be set to the issuer URL of the authorization server.

aud MUST identify the resource server receiving the token introspection response.

iat MUST be set to the time when the introspection response was created by the authorization server.

token_introspection A JSON object containing the members of the token introspection response, as specified in the "OAuth Token Introspection Response" registry established by [RFC7662] as well as other members. The separation of the introspection members into a dedicated containing JWT claim is intended to prevent conflict and confusion with top-level JWT claims that may bear the same name.

If the access token is invalid, expired, revoked, or is not intended for the calling resource server (audience), the authorization server MUST set the value of the "active" member in the "token_introspection" claim to "false" and other members MUST NOT be included. Otherwise, the "active" member is set to "true".

If possible, the AS MUST narrow down the "scope" value to the scopes relevant to the particular RS.

Claims from the "JSON Web Token Claims" registry that are commonly used in [OpenID.Core] and can be applied to the resource owner MAY be included as members in the "token_introspection" claim. They can serve to convey the privileges delegated to the client, to identify the resource owner as a natural person or to provide a required contact detail, such as an e-Mail address or phone number. When transmitting such claims the AS acts as an identity provider in regard to the RS. The AS determines based on its RS-specific policy what claims about the resource owner to return in the token introspection response.

The AS MUST ensure the release of any privacy-sensitive data is legally based.

Further content of the introspection response is determined by the RS-specific policy at the AS.

The JWT MAY include other claims, including those from the "JSON Web Token Claims" registry established by [RFC7519]. The JWT SHOULD NOT include the "sub" and "exp" claims as an additional prevention against misuse of the JWT as an access token (see Section 8.1).

Note: Although the JWT format is widely used as an access token format, the JWT returned in the introspection response is not an alternative representation of the introspected access token and is not intended to be used as an access token.

This specification registers the "application/token-introspection+jwt" media type, which is used as value of the "typ" ("type") header parameter of the JWT to indicate that the payload is a token introspection response.

The JWT is cryptographically secured as specified in [RFC7662].

Depending on the specific resource server policy the JWT is either signed, or signed and encrypted. If the JWT is signed and encrypted it MUST be a Nested JWT, as defined in JWT [RFC7519].

Note: If the resource server policy requires a signed and encrypted response and the authorization server receives an unauthenticated request containing an "Accept" header with content type other than "application/token-introspection+jwt", it MUST refuse to serve the request and return an HTTP status code 400. This is done to prevent downgrading attacks to obtain token data intended for release to legitimate recipients only (see Section 8.2).

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

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

eyJraWQiOiJ3RzZEIiwidHlwIjoidG9rZW4taW50cm9zcGVjdGlvbitqd3QiLCJhbGc
iOiJSUzI1NiJ9.eyJpc3MiOiJodHRwczovL2FzLmV4YW1wbGUuY29tLyIsImF1ZCI6I
mh0dHBzOi8vcnMuZXhhbXBsZS5jb20vcmVzb3VyY2UiLCJpYXQiOjE1MTQ3OTc4OTIs
InRva2VuX2ludHJvc3BlY3Rpb24iOnsiYWN0aXZlIjp0cnVlLCJpc3MiOiJodHRwczo
vL2FzLmV4YW1wbGUuY29tLyIsImF1ZCI6Imh0dHBzOi8vcnMuZXhhbXBsZS5jb20vcm
Vzb3VyY2UiLCJpYXQiOjE1MTQ3OTc4MjIsImV4cCI6MTUxNDc5Nzk0MiwiY2xpZW50X
2lkIjoicGFpQjJnb28wYSIsInNjb3BlIjoicmVhZHdyaXRlZG9scGhpbiIsInN1YiI6
Ilo1TzN1cFBDODhRckFqeDAwZGlzIiwiYmlydGhkYXRlIjoiMTk4Mi0wMi0wMSIsImd
pdmVuX25hbWUiOiJKb2huIiwiZmFtaWx5X25hbWUiOiJEb2UiLCJqdGkiOiJ0MUZvQ0
NhWmQ0WHY0T1JKVVdWVWVUWmZzS2hXMzBDUUNyV0REandYeTZ3In19.przJMU5GhmNz
vwtt1Sr-xa9xTkpiAg5IshbQsRiRVP_7eGR1GHYrNwQh84kxOkHCyje2g5WSRcYosGE
VIiC-eoPJJ-qBwqwSlgx9JEeCDw2W5DjrblOI_N0Jvsq_dUeOyoWVMqlOydOBhKNY0s
mBrI4NZvEExucOm9WUJXMuJtvq1gBes-0go5j4TEv9sOP9uu81gqWTr_LOo6pgT0tFF
yZfWC4kbXPXiQ2YT6mxCiQRRNM-l9cBdF6Jx6IOrsfFhBuYdYQ_mlL19HgDDOFaleyq
mru6lKlASOsaE8dmLSeKcX91FbG79FKN8un24iwIDCbKT9xlUFl54xWVShNDFA
```

The example response JWT header contains the following JSON document:

```
{
  "typ": "token-introspection+jwt",
  "alg": "RS256"
  "kid": "wG6D"
}
```

The example response JWT payload contains the following JSON document:

```
{
  "iss":"https://as.example.com/",
  "aud":"https://rs.example.com/resource",
  "iat":1514797892,
  "token_introspection":
     {
        "active":true,
        "iss":"https://as.example.com/",
        "aud":"https://rs.example.com/resource",
        "iat":1514797822,
        "exp":1514797942,
        "client_id":"paiB2goo0a",
        "scope":"read write dolphin",
        "sub":"Z5O3upPC88QrAjx00dis",
        "birthdate":"1982-02-01",
        "given_name":"John",
        "family_name":"Doe",
        "jti":"t1FoCCaZd4Xv4ORJUWVUeTZfsKhW30CQCrWDDjwXy6w"
     }
}
```

6. Client Metadata

The authorization server determines the algorithm 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 [RFC7591] with the resource server acting as a client, as specified below.

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 OPTIONAL. JWS [RFC7515] algorithm ("alg" value) as defined in JWA [RFC7518] for signing

introspection responses. If this is specified, the response will be signed using JWS and the configured algorithm. The default, if omitted, is "RS256".

introspection_encrypted_response_alg OPTIONAL. JWE [RFC7516] algorithm ("alg" value) as defined in JWA [RFC7518] for content key encryption. If this is specified, the response will be encrypted using JWE and the configured content encryption algorithm ("introspection_encrypted_response_enc"). The default, if omitted, is that no encryption is performed. 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].

introspection_encrypted_response_enc OPTIONAL. JWE [RFC7516] algorithm ("enc" value) as defined in JWA [RFC7518] for content encryption of introspection responses. The default, if omitted, is "A128CBC-HS256". Note: This parameter MUST NOT be specified without setting "introspection_encrypted_response_alg".

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

7. Authorization Server Metadata

Authorization servers SHOULD publish the supported algorithms for signing and encrypting the JWT of an introspection response by utilizing OAuth 2.0 Authorization Server Metadata [RFC8414] parameters. Resource servers use this data to parametrize their client registration requests.

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) as defined in 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) as defined in JWA [RFC7518] supported by the introspection endpoint to encrypt the content encryption key for introspection responses (content key encryption).

introspection_encryption_enc_values_supported OPTIONAL. JSON array containing a list of the JWE [RFC7516] encryption algorithms

("enc" values) as defined in JWA [RFC7518] supported by the introspection endpoint to encrypt the response (content encryption).

8. Security Considerations

8.1. Cross-JWT Confusion

The "iss" and potentially the "aud" claim of a token introspection JWT can resemble those of a JWT-encoded access token. An attacker could try to exploit this and pass a JWT token introspection response as an access token to the resource server. The "typ" ("type") JWT header "token-introspection+jwt" and the encapsulation of the token introspection members such as "sub" and "scope" in the "token_introspection" claim is intended to prevent such substitution attacks. Resource servers MUST therefore check the "typ" JWT header value of received JWT-encoded access tokens and ensure all minimally required claims for a valid access token are present.

Resource servers MUST additionally apply the countermeasures against replay as described in [I-D.ietf-oauth-security-topics], section 3.2.

JWT Confusion and other attacks involving JWTs are discussed in [I-D.ietf-oauth-jwt-bcp].

8.2. Token Data Leakage

The authorization server MUST use Transport Layer Security (TLS) 1.2 (or higher) per BCP 195 [RFC7525] in order to prevent token data leakage.

To prevent introspection of leaked tokens and to present an additional security layer against token guessing attacks the authorization server MAY require all requests to the token introspection endpoint to be authenticated. As an alternative or as an addition to the authentication, the intended recipients MAY be set up for encrypted responses.

In the latter case, confidentiality is ensured by the fact that only the legitimate recipient is able to decrypt the response. An attacker could try to circumvent this measure by requesting a plain JSON response, using an "Accept" header with the content type set to, for example, "application/json" instead of "application/token-introspection+jwt". To prevent this attack the authorization server MUST NOT serve requests with a content type other than "application/token-introspection+jwt" if the resource server is set up to receive encrypted responses (see also Section 5).

8.3. Keeping Token Data Confidential from OAuth Clients

Authorization servers with a policy that requires token data to be kept confidential from OAuth clients must require all requests to the token introspection endpoint to be authenticated. As an alternative or as an addition to the authentication, the intended recipients may be set up for encrypted responses.

8.4. Logging and Audit of Introspection Activity

Authorization servers with a policy that requires token introspection activity to be logged and audited must require all requests to the token introspection endpoint to be authenticated.

9. Privacy Considerations

The token introspection response can be used to transfer personal identifiable information from the AS to the RS. The AS MUST ensure a legal basis exists for the data transfer before any data is released to a particular RS. The way the legal basis is established might vary among jurisdictions and MUST consider the legal entities involved.

For example, the classical way to establish the legal basis is by explicit user consent gathered from the resource owner by the AS during the authorization flow.

It is also possible that the legal basis is established out of band, e.g. in an explicit contract or by the client gathering the resource owner's consent.

If the AS and the RS belong to the same legal entity (1st party scenario), there is potentially no need for an explicit user consent but the terms of service and policy of the respective service provider MUST be enforced at all times.

In any case, the AS MUST ensure that the scope of the legal basis is enforced throughout the whole process. The AS MUST retain the scope of the legal basis with the access token, e.g. in the scope value, and the AS MUST determine the data a resource server is allowed to receive based on the resource server's identity and suitable token data, e.g. the scope value.

10. Acknowledgements

We would like to thank Petteri Stenius, Neil Madden, Filip Skokan, Tony Nadalin, Remco Schaar, Justin Richer and Takahiko Kawasaki for their valuable feedback.

11. IANA Considerations

11.1. 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]:

11.1.1. 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 6 of [[this specification]]
- Client Metadata Name: "introspection_encrypted_response_alg"
- Client Metadata Description: String value specifying the desired introspection response content key encryption algorithm (alg value).
- Change Controller: IESG
- Specification Document(s): Section 6 of [[this specification]]
- Client Metadata Name: "introspection_encrypted_response_enc"
- Client Metadata Description: String value specifying the desired introspection response content encryption algorithm (enc value).
- Change Controller: IESG
- Specification Document(s): Section 6 of [[this specification]]

11.2. OAuth Authorization Server Metadata Registration

This specification requests registration of the following values in the IANA "OAuth Authorization Server Metadata" registry [IANA.OAuth.Parameters] established by [RFC8414].

11.2.1. 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 7 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 content key encryption (alg value).
- Change Controller: IESG
- Specification Document(s): Section 7 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 content encryption (enc value).
- Change Controller: IESG
- Specification Document(s): Section 7 of [[this specification]]

11.3. Media Type Registration

This section registers the "application/token-introspection+jwt" media type in the "Media Types" registry [IANA.MediaTypes] in the manner described in [RFC6838], which can be used to indicate that the content is a token introspection response in JWT format.

11.3.1. Registry Contents

- Type name: application
- Subtype name: token-introspection+jwt
- Required parameters: N/A
- Optional parameters: N/A

- Encoding considerations: binary; A token introspection response is a JWT; JWT values are encoded as a series of base64url-encoded values (with trailing '=' characters removed), some of which may be the empty string, separated by period ('.') characters.

- Security considerations: See Section 7 of this specification

- Interoperability considerations: N/A

- Published specification: Section 4 of this specification

- Applications that use this media type: Applications that produce and consume OAuth Token Introspection Responses in JWT format

- Fragment identifier considerations: N/A

- Additional information:

  * Magic number(s): N/A

  * File extension(s): N/A

  * Macintosh file type code(s): N/A

- Person & email address to contact for further information: Torsten Lodderstedt, torsten@lodderstedt.net

- Intended usage: COMMON

- Restrictions on usage: none

- Author: Torsten Lodderstedt, torsten@lodderstedt.net

- Change controller: IESG

- Provisional registration? No

11.4. JWT Claim Registration

This section registers the "token_introspection" claim in the JSON Web Token (JWT) IANA registry [IANA.JWT] in the manner described in [RFC7519].

11.4.1. Registry Contents

- Claim name: token_introspection

- Claim description: Token introspection response

- Change Controller: IESG

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

12. References

12.1. Normative References

[I-D.ietf-oauth-jwt-bcp]
Sheffer, Y., Hardt, D., and M. Jones, "JSON Web Token Best Current Practices", draft-ietf-oauth-jwt-bcp-06 (work in progress), June 2019.

[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-13 (work in progress), July 2019.

[IANA.JWT]
IANA, "JSON Web Token (JWT)", <https://www.iana.org/assignments/jwt/jwt.xhtml>.

[IANA.MediaTypes]
IANA, "Media Types", <http://www.iana.org/assignments/media-types>.

[OpenID.Core]
Sakimura, N., Bradley, J., Jones, M., Medeiros, B. D., and C. Mortimore, "OpenID Connect Core 1.0 incorporating errata set 1", Nov 2014, <http://openid.net/specs/openid-connect-core-1_0.html>.

[OpenID.Registration]
Sakimura, N., Bradley, J., and M. Jones, "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>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[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>.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <https://www.rfc-editor.org/info/rfc7525>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8414] Jones, M., Sakimura, N., and J. Bradley, "OAuth 2.0 Authorization Server Metadata", RFC 8414, DOI 10.17487/RFC8414, June 2018, <https://www.rfc-editor.org/info/rfc8414>.

12.2. 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]]

-09

- changes the Accept and Content-Type HTTP headers from "application/json" to "application/token-introspection+jwt" so they match the registered media type
- moves the token introspection response members into a JSON object claim named "token_introspection" to provide isolation from the top-level JWT-specific claims
- "iss", "aud" and "iat" MUST be present as top-level JWT claims
- the "sub" and "exp" claims SHOULD NOT be used as top-level JWT claims as additional prevention against JWT access token substitution attacks

-08

- made difference between introspected access token and introspection response clearer
- defined semantics of JWT claims overlapping between introspected access token and introspection response as JWT
- added section about RS management
- added text about user claims including a privacy considerations section
- removed registration of OpenID Connect claims to "Token Introspection Response" registry and refer to "JWT Claims" registry instead
- added registration of "application/token-introspection+jwt" media type as type identifier of token introspection responses in JWT format
- more changed to incorporate IESG review feedback

-07

- fixed wrong description of "locale"
- added references for ISO and ITU specifications

-06

- replaced reference to RFC 7159 with reference to RFC 8259

-05

- improved wording for TLS requirement
- added RFC 2119 boilerplate
- fixed and updated some references

-04

- reworked definition of parameters in section 4
- added text on data minimization to security considerations section
- added statement regarding TLS to security considerations section

-03

- added registration for OpenID Connect Standard Claims to OAuth Token Introspection Response registry

-02

- updated references

-01

- adapted wording to preclude any accept header except "application/jwt" if encrypted responses are required
- use registered alg value RS256 for default signing algorithm
- added text on claims in the token introspection response

-00

- initial version of the WG draft
- defined default signing algorithm
- changed behavior in case resource server is set up for encryption
- Added text on token data leakage prevention to the security considerations

- moved Security Considerations section forward

WG draft

-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