Network Working Group
Internet-Draft
Intended status: Experimental
Expires: January 19, 2017

J. Bradley, Ed.
Ping
J. Richer

July 18, 2016

Stateless Client Identifier for OAuth 2
draft-bradley-oauth-stateless-client-id-03

Abstract

This draft provides a method for communicating information about an OAuth client through its client identifier allowing for fully stateless operation.

Requirements Language

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].

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 January 19, 2017.

Copyright Notice

Table of Contents

1. Introduction

In the OAuth 2.0 Authorization protocol, the Client must provide a Client Identifier that the Authorization Server recognizes. Additionally, an Autorization Server needs to know about a client's details, such as its name and redirect URIs. Traditionally, this is handled through a registration process, which may be either manual or automated, where the authorization server maintains a stateful relationship between the Client Identifier and its associated metadata. This draft proposes a mechanism whereby the essential metadata can be encoded into the Client Identifier itself, signed by the issuer, and validated by the authorization server, thus allowing the authorization server to be stateless in regard to client information.

2. Stateless Client Identifier

The stateless client identifier consists of a JWT [RFC7519], optionally signed with JWS [RFC7515], whose payload contains claims as defined here.

iss REQUIRED. URL identifying the party that issued this client identifier.

sub REQUIRED. Identifier of the client, locally unique to the issuer.

cnf OPTIONAL. JWT Confirmation claim [RFC7800] providing information for the token _endpoint to validate the presenter of the client_id

iat OPTIONAL. Timestamp of when this client identifier was issued.

exp OPTIONAL. Timestamp of when this client identifier will expire.

kid RECOMMENDED if signed. Identifier of the key used to sign this client identifier at the issuer.

reg REQUIRED. JSON Object containing a set of metadata claims of client information such as its redirect URIs, display name, and other items as defined in Dynamic Client Registration [RFC7591] and its extensions.

The issuer SHOULD sign the JWT with JWS in such a way that the signature can be verified by the authorization server.

The issuer MAY encrypt the JWT with JWE [RFC7516].

3. Validating the Stateless Client Identifier

Upon receiving a stateless client identifier at either the authorization endpoint or the token endpoint, the authorization server parses it as a JWT. It first checks the iss field to determine if it trusts identifiers issued by the party represented. It then verifies the signature if the JWT (if signed) using JWS. The key used to sign the JWT MAY be indicated by the kid field. The authorization server MAY use other means to validate the JWT and determine its authenticity.

The authorization server then reads the fields inside the reg claim and uses these to configure the user experience and security parameters of the authorization.

4. Obtaining a Stateless Client Identifier

The client identifier is intended to be opaque to the client, and as such a stateless client identifier is intended to be obtained and used in exactly the same way as a stateful client identifer would be for any OAuth client.

- Manual registration: a developer uses an out-of-band adminstrative process to generate the client identifier and related credentials.

- Dynamic registration: a developer or client uses the process described in [Dyn Reg] to generate the client identifier and related credentials.

- Self assertion: a developer or client generates the client identifier on their own, often signing it with their own public key.

It is completely up to the purview of particular authorization servers which generation methods, and which client identifiers, they will accept.

5. IANA Considerations

[maybe we register the "reg" claim above?]

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

Since many OAuth systems assume that a change in the client identifier indicates a change in the client itself, systems using stateless client identifiers SHOULD NOT allow clients to update their information post registration.

Since the client identifier is passed through the browser to the authorization endpoint, it MUST NOT contain any sensitive information. Additionally, as in standard OAuth, posession of the client identifier itself MUST NOT be assumed to be sufficient authentication [in many cases? except implicit mode?].

7. Acknowledgements

8. Normative References

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

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

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

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <http://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, <http://www.rfc-editor.org/info/rfc7591>.

[RFC7800] Jones, M., Bradley, J., and H. Tschofenig, "Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs)", RFC 7800, DOI 10.17487/RFC7800, April 2016, <http://www.rfc-editor.org/info/rfc7800>.

Appendix A. Document History

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

-03

- Added reference to RFC7800 and using the cnf claim in a clinet_id (I am thinking that if the dynamic client registration endpoint is on the same host as the token_endpoint, you could use the token binding ID as the cnf method.

-02

- Added reference to RFC7591

-01

- Added reference to RFC for JOSE

-00

- Wrote the first draft.

Authors' Addresses

John Bradley (editor)
Ping Identity

Email: ve7jtb@ve7jtb.com
URI: http://www.thread-safe.com/

Justin Richer