Open Authentication Protocol
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

A. Parecki
Okta

February 22, 2021

OAuth 2.0 Client Intermediary Metadata
draft-parecki-oauth-client-intermediary-metadata-03

Abstract

This specification defines a mechanism for including information about additional parties involved in an OAuth transaction by adding a new section to the OAuth 2.0 Dynamic Client Registration request, as well as requires that authorization servers surface this information to users during an OAuth transaction.

1. Introduction

In some applications of OAuth, there may be multiple legal entities which have access to or process data retrieved by an OAuth client. In the traditional OAuth model, a "client_id" represents only a single application, and so the OAuth consent screen lists just one third party: the OAuth client.

In this situation, in order to comply with various local laws and regulations, the user needs to be informed by the authorization server of the list of entities that will have access to their data after authorizing the client.

The existing Dynamic Client Registration ([RFC7591]) specification lacks a mechanism for communicating a list of additional parties that may have access to the user's data.

This specification extends [RFC7591] and [RFC7592] to define a mechanism for including information about the additional parties involved in an OAuth transaction by including information about the additional intermediaries into the Dynamic Client Registration request. This specification also defines requirements of the OAuth authorization server to present this information about the additional parties in the OAuth consent screen during an OAuth transaction.

2. Notational 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 [RFC2119].

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

3. Terminology

In addition to the terms defined in referenced specifications, this document uses the following terms:

"OAuth": In this document, "OAuth" refers to OAuth 2.0, [RFC6749].

"Client": "Client" has the same definition as in OAuth 2.0, but is worth pointing out that the client in this context may be operated by a different legal entity than is described by the client name.

"Intermediary": One or more entities that the user's data will pass through or be shared with by using the OAuth client. This

information is voluntarily provided by the OAuth client, and is typically enforced by a business relationship between the organization providing the Client and the organization providing the Resource Server.

4. Client Intermediary Metadata

Registered client intermediaries have a set of metadata values associated with the client identifier of the client that represents them in the OAuth transaction, such as a user-visible name, logo, and URL.

Like the OAuth client metadata defined in [RFC7591] and [RFC7592], these metadata values are used in the following ways:

- as input values to registration and update requests, and
- as output values in registration responses.

These values are used by the authorization server when displaying the OAuth consent screen to the end user, to inform them of all the additional parties that will be handling the user's data upon approval.

The following metadata fields are defined by this specification. The implementation and use of the fields is OPTIONAL unless stated otherwise. All data member types (strings, arrays, numbers) are defined in terms of their JSON ([RFC7159]) representations.

Some fields are expected to be displayed in the OAuth consent UI and are designated accordingly.

"name"

REQUIRED. A human-readable name of intermediary party. Authorization servers MUST display this field to the end user on the OAuth consent screen.

"description"

OPTIONAL. A human-readable description of the intermediary. This is not intended to be displayed in the OAuth consent screen.

"uri"

A URL string of a web page providing information about the intermediary. If present, the authorization server SHOULD display this URL to the end user in a clickable fashion. It is RECOMMENDED

that clients always send this field. The value of this field MUST point to a valid web page.

"logo_uri"

A URL string that references a logo for this intermediary. If present, the authorization server SHOULD display this image to the end user in the OAuth consent screen. The value of this field MUST be a valid image file.

"contacts"

Array of strings representing ways to contact people responsible for this intermediary, typically email addresses or phone numbers. The authorization server MAY display these to the end user in the OAuth consent screen. See Section 6 of [RFC7591] for information on Privacy Considerations.

5. Client Registration Endpoint

The client registration endpoint is described in Section 3 of [RFC7591].

Since this specification provides a mechanism for a client to assert information about additional parties other than itself, the registration endpoint MUST be protected by an OAuth 2.0 access token obtained by the client. The method by which the initial access token is obtained by the client or developer is out of scope of this specification, but is likely to be obtained using the client credentials grant or manual out-of-band registration.

5.1. Client Registration Request

This specification extends the client registration request defined in [RFC7591].

This operation registers a combination of client and one or more intermediaries with an authorization server. The authorization server assigns a unique client identifier (and optionally a client secret) that represents the combination of all the entities described in the registration request.

To register, the client or developer sends an HTTP POST as described in Section 3.1 of [RFC7591], with an additional property named "intermediaries" with a JSON array of objects of each intermediary's registration information. The properties of the object are described above in Section 4.

For example, the client could send the following registration request to the client registration endpoint using its OAuth 2.0 access token it has previously obtained using the client credentials grant.

The following is a non-normative example request:

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: server.example.com
Authorization: Bearer 8IGFGXKXZBV5LL38Y3X1

{
  "client_name": "User-Recognizable App Name",
  "redirect_uris": [
    "https://client.example.org/callback"
  ],
  "client_uri": "https://example.net/",
  "logo_uri": "https://example.net/logo.png",
  "contacts": [
    "support@example.net"
  ],
  "intermediaries": [
    {
      "name": "Partner App Name",
      "description": "An application that may also receive
        this user's data when the user authorizes the client",
      "uri": "https://partner.example/",
      "logo_uri": "https://partner.example/logo.png",
      "contacts": [
        "support@partner.example"
      ]
    }
  ]
}
```

5.2. Client Registration Response

This specification extends the client information response defined in [RFC7591] and [RFC7592].

Upon a successful registration request, the authorization server returns a client identifier for the combination of the client and any intermediaries specified in the request.

In addition to the response fields defined in Section 3.2 of [RFC7591] and Section 3 of [RFC7592], the response MUST also contain all registered metadata about the intermediaries. The authorization

server MAY reject or replace any of the requested metadata values submitted during the registration and substitute them with suitable values.

The following is a non-normative example response of a successful registration:

```
 HTTP/1.1 201 Created
 Content-Type: application/json
 Cache-Control: no-store
 Pragma: no-cache

 {
   "client_id": "V8tvEkZWhDAdxSaKGUJZ",
   "client_secret": "SpsuwZIxnp8bBEhp5sk1EKiIKTZ4X4DKU",
   "grant_types": ["authorization_code", "refresh_token"],
   "token_endpoint_auth_method": "client_secret_basic",
   "registration_client_uri": "https://server.example.com/client/
tmzaAMkyWlH3",
   "registration_access_token": "MphaAqDaZT86C93ENWRZcf3dfU2dW6POASo8dFXa",
   "client_name": "User-Recognizable App Name",
   "client_uri": "https://example.net/",
   "redirect_uris": [
     "https://client.example.org/callback"
   ],
   "contacts": [
     "support@example.net"
   ],
   "intermediaries": [
     {
       "name": "Partner App Name",
       "description": "An application that may also receive
         this user's data when the user authorizes the client",
       "uri": "https://partner.example/",
       "logo_uri": "https://partner.example/logo.png",
       "contacts": [
         "support@partner.example"
       ]
     }
   ]
 }
```

The "registration_client_uri" and "registration_access_token" properties are required in order to support updating and deleting this client as described in [RFC7592].

5.3. Client Read Request

This specification extends the client read request defined in [RFC7592] to include the additional metadata properties in the response that describe the intermediaries. No additional behavior is prescribed by this specification.

5.4. Client Update Request

This specification extends the client update request defined in [RFC7592] to be able to update the additional metadata properties that describe the intermediaries.

The additional properties are provided in the update request in the same format as in the initial registration request.

Since these values were asserted by the client in the initial registration, there is no need to prescribe any additional security model around the ability to update them, even though these represent additional parties.

5.5. Client Delete Request

No new behavior is prescribed for delete requests beyond that defined in [RFC7592].

6. Providing Intermediaty Details in the Authorization Request

When the authorization server begins a request from an OAuth client identifier that has been registered with additional intermediary information, it MUST display the additional parties in the consent UI visible to the end user.

The authorization server chooses how best to display the additional information, but it MUST include at least the name of the intermediaries and client, and SHOULD include the logo of each as well.

7. Security Considerations

As this extends [RFC7591], all security considerations from that draft apply here as well.

Specifically, if the authorization server supports open client registration without any authentication, it must be extremely careful with any URLs received in the registration request such as "logo_uri", "tos_uri", and "uri", as these values may be displayed to end users. [RFC7591] recommends requiring that these URIs have a

matching host and scheme as the defined "redirect_uri"s, and that they are resolvable URIs. See section 5 of [RFC7591] for more details.

8. Acknowledgements

The author would like to thank Ryan Christiansen and Preston McFarland for their initial contributions of the concepts behind this specification. The author would also like to thank Don Cardinal, Ryan Christiansen and Preston McFarland for their reviews of this specification, as well as Anil Mahalaha and other members of the Financial Data Exchange working group. Additionally the work of the OAuth Working Group on the referenced and related specifications that this specification builds upon is much appreciated.

9. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/info/rfc6749>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[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>.

[RFC7592] Richer, J., Ed., Jones, M., Bradley, J., and M. Machulak, "OAuth 2.0 Dynamic Client Registration Management Protocol", RFC 7592, DOI 10.17487/RFC7592, July 2015, <https://www.rfc-editor.org/info/rfc7592>.

Author's Address

Aaron Parecki
Okta

Email: aaron@parecki.com
URI: https://aaronparecki.com