OAuth Identity and Authorization
Chaining Across Domains

IETF 119 OAuth Working Group Meeting

Arndt Schwenkschuster (Microsoft)
Pieter Kasselman (Microsoft)
Kelley Burgin, (MITRE)
Mike Jenkins (NSA-CSS)
Brian Campbell (Ping)
Agenda

• The challenge of Identity Chaining
• The approach
• What’s in the draft
• Changes since IETF 118
• Next Steps
Why Identity Chaining Across Trust Domains

**Transaction Tokens**
- Who was the Resource Owner?
- What authorization did they grant?
- What other entities were involved?
- What authorization did they have?

Image Courtesy Justin Richer (justin@bspk.io)
Why Identity Chaining Across Trust Domains

Client

Authorization Server Domain 1

Resource Owner

Trust Domain 1

Gateway

Foo

Bar

Trust Domain 2

Gateway

Baz

Qux

Image Courtesy Justin Richer (justin@bspk.io)
Why Identity Chaining Across Trust Domains

- Who was the Resource Owner?
- What authorization did they grant?
- What other entities were involved?
- What authorization did they have?

Image Courtesy Justin Richer (justin@bspk.io)
Why Identity Chaining Across Trust Domains

Client -> Authorization Server Domain 1 -> Resource Owner -> Trust Domain 1

Client -> Authorization Server Domain 2 -> Resource Owner -> Trust Domain 2

Image Courtesy Justin Richer (justin@bspk.io)
Proposal Concepts
Getting an Authorization Grant for another Trust Domain

1. Client in Domain 1 exchanges a token with the AS in Trust Domain 1 to get an authorization grant for the AS in Trust Domain 2

2. Client use the authorization grant with the AS in Trust Domain 2 to get an access token

3. Client presents the access token to the Resource server in Trust Domain 2

Related standards:
- Token Exchange (RFC 8693)
-Assertion Framework (RFC 7521)
Generic Cross-Domain Identity Chaining

1. Token Exchange – Obtain Authorization Grant
2. Assertion Framework – Get Access Token
3. Present Access Token

Trust Domain 1
- Authorization Server Domain 1
- Gateway
- Foo
- Bar

Trust Domain 2
- Authorization Server Domain 2
- Gateway
- Baz
- Qux
Resource Server as Client

1. Authorization Server Domain 1
   - Token Exchange – Obtain Authorization Grant
2. Gateway Foo
   - Present Access Token
3. Client 2
   - Get Access Token
4. Gateway Bar
5. Gateway Baz
6. Gateway Qux

Assertion Framework – Get Access Token
Authorization Server Domain 2

Trust Domain 1
Trust Domain 2
Authorization Server as Client

1. Token Exchange – Request access token for domain 2

2. Assertion Framework – Get Access Token

3. Present Access Token

Token Exchange – Obtain Authorization Grant

Authorization Server Domain 1
Client 2

Trust Domain 1

Gateway
Foo
Bar

Trust Domain 2

Gateway
Baz
Qux

Authorization Server Domain 2

What’s in the Draft

Section 2.1 Overview: Generic Identity and Authorization Chaining Across Domains
Appendix B.1: Resource Server acting as Client
Appendix B.2: Authorization Server acting as Client
2.3. Token Exchange

The client performs token exchange as defined in [RFC8693] with the authorization server for its own domain (e.g., Domain A) in order to obtain a JWT authorization grant that can be used with the authorization server of a different domain (e.g., Domain B) as specified in section 1.3 of [RFC6749].

2.3.1. Token Exchange Request

The parameters described in section 2.1 of [RFC8693] apply here with the following restrictions:

requested_token_type
  - OPTIONAL according to [RFC8693]. In the context of this specification this parameter SHOULD NOT be used.

scope
  - OPTIONAL. Additional scopes to indicate scopes included in the returned JWT authorization grant. See Claims transcription (Section 2.5).

resource
  - REQUIRED if audience is not set. URI of authorization server of targeting domain (domain B).

audience
  - REQUIRED if resource is not set. Well known/logical name of authorization server of targeting domain (domain B).

2.3.2. Processing rules

- If the request itself is not valid or if the given resource or audience are unknown, or are unacceptable based on policy, the authorization server MUST deny the request.
- The authorization server MAY add, remove or change claims. See Claims transcription (Section 2.5).
2.4.1. Access Token Request

The authorization grant is a JWT bearer token, which the client uses to request an access token as described in the JWT Profile for OAuth 2.0 Client Authentication and Authorization Grants [RFC7523]. For the purpose of this specification the following descriptions apply:

grant_type

REQUIRED. As defined in Section 2.1 of [RFC7523] the value `urn:ietf:params:oauth:grant-type:jwt-bearer` indicates the request is a JWT bearer assertion authorization grant.

assertion

REQUIRED. Authorization grant returned by the token exchange (access_token response).

scope

OPTIONAL.

The client MAY indicate the audience it is trying to access through the scope parameter or the resource parameter defined in [RFC8707].

2.4.2. Processing rules

The authorization server MUST validate the JWT authorization grant as specified in Sections 3 and 3.1 of [RFC7523]. The following processing rules also apply:

- The "aud" claim MUST identify the Authorization Server as a valid intended audience of the assertion using either the token endpoint as described Section 3 [RFC7523] or the issuer identifier as defined in Section 2 of [RFC8414].
- The authorization server SHOULD deny the request if it is not able to identify the subject.
- Due to policy the request MAY be denied (for instance if the federation from domain A is not allowed).
2.5. Claims transcription

Authorization servers MAY transcribe claims when either producing JWT authorization grants in the token exchange flow or access tokens in the assertion flow.

- Transcribing the subject identifier: Subject identifier can differ between the parties involved. For instance: A user is known at domain A by "johndoe@a.org" but in domain B by "doe.john@b.org". The mapping from one identifier to the other MAY either happen in the token exchange step and the updated identifier is reflected in returned JWT authorization grant or in the assertion step where the updated identifier would be reflected in the access token. To support this both authorization servers MAY add, change or remove claims as described above.
- Selective disclosure: Authorization servers MAY remove or hide certain claims due to privacy requirements or reduced trust towards the targeting trust domain. To hide and enclose claims [I-D.ietf-oauth-selective-disclosure-jwt] MAY be used.
- Controlling scope: Clients MAY use the scope parameter to control transcribed claims (e.g. downscooping). Authorization Servers SHOULD verify that requested scopes are not higher privileged than the scopes of presented subject_token.
- Including JWT authorization grant claims: The authorization server performing the assertion flow MAY leverage claims from the presented JWT authorization grant and include them in the returned access token. The populated claims SHOULD be namespace or validated to prevent the injection of invalid claims.

The representation of transcribed claims and their format is not defined in this specification.
Changes since IETF 118
Changes in -01

- Changed name to "OAuth Identity and Authorization Chaining Across Domains"
- Limit the authorization grant format to RFC7523 JWT
- Use more specific term "JWT authorization grant"
- Renamed section headers to be more explicit
- Moved use cases to appendix and add continuous integration use case
- Minor example fixes
- Editorial fixes

https://github.com/oauth-wg/oauth-identity-chaining/issues
Next Steps
<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Status</th>
<th>Mentioned by</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Figure 6 step naming misalignment (off-by-one error)</strong></td>
<td>#88 opened last week by dhs-aws</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Add sender constraining mechanisms</strong></td>
<td>#85 opened last week by kburgin3</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Recommended media type for JWT Authorization Grant</strong></td>
<td>#85 opened 2 weeks ago by aaronpk</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>How does the AS know it should generate a JWT Authorization Grant formatted token?</strong></td>
<td>#84 opened 2 weeks ago by gffletch</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>relax recommendation against requested_token_type use</strong></td>
<td>#82 opened 3 weeks ago by bc-pi</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Should we allow identity chaining with DPoP tokens?</strong></td>
<td>#79 opened last month by arndt-s</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Describe or give examples of what kinds of tokens a client would exchange</strong></td>
<td>#75 opened on Feb 7 by aaronpk</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Add use case to the appendix</strong></td>
<td>#68 opened on Dec 7, 2023 by PieterKas</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions?