OAuth 2.1

Interim Meeting • Oct 13, 2021

`draft -04`

Aaron Parecki, Dick Hardt, Torsten Lodderstedt

Changes Since draft -02

- Incorporated errata on RFC6749 (Thanks adeinega)
- Removed “Pragma” header (Thanks adeinega and bc)
- Editorial clarifications based on feedback and PRs
  - Replaced use of “user-agent” (the HTTP header) with “user agent” (Thanks ioggstream)
  - Fixed references to RFC7230 vs RFC7231
  - Lots more! Thanks Justin and Vittorio!
- Rearranged section 4 to be about all token endpoint requests rather than “obtaining authorization”
  - Including authorization code, refresh token, client credentials
- Moved some normative text from security considerations inline in the main document
- Added explicit mention of avoiding sending access tokens in query strings

Planned Changes for -05

- #70 Finish incorporating feedback from Justin and Vittorio (Sections 8-13)
- #64 Finish moving normative language from security considerations inline in the doc
- #45 Add non-normative note about removal of Implicit flow as discussed in previous Interim

Issues for Discussion

#45 Referencing OpenID Connect Implicit Flows

Proposed new text to add to section 10

10.1. Removal of the OAuth 2.0 Implicit grant

The OAuth 2.0 Implicit grant is omitted from OAuth 2.1 as it was deprecated in [I-D.ietf-oauth-security-topics].

The intent of removing the Implicit grant is to no longer issue access tokens in the authorization response, as such tokens are vulnerable to leakage and injection, and are unable to be sender-constrained to a client. This behavior was indicated by clients using the response_type=token parameter. This value for the response_type parameter is no longer defined in OAuth 2.1.

Removal of response_type=token does not have an effect on other extension response types returning other artifacts from the authorization endpoint, for example, response_type=id_token defined by [OpenID].

#46 `iss` response parameter

The Security BCP will be recommending the use of the `iss` response parameter to defend against AS mixup attacks

Proposal:

- We should add this to OAuth 2.1 despite it being a relatively late addition to the Security BCP

Rationale:

- Add a solid and simple mix-up prevention to OAuth 2.1 for clients interacting with multiple ASs

#40 Require HTTPS redirect URLs

Current:

- The redirection endpoint SHOULD require the use of TLS …
- … If TLS is not available, the authorization server SHOULD warn the resource owner about the insecure endpoint prior to redirection

Proposed:

- Require TLS except for loopback and custom URL schemes

#92 Drop complex authorization code replay mitigations in favor of PKCE

There are a number of authorization code replay mitigations in 6749 that are not strictly necessary thanks to PKCE. Additionally, some of the mitigation techniques are quite burdensome to implement.

- #65 Single-use authorization codes
  - Some implementations do not enforce this today
- #82 Should authorization codes be invalidated if used unsuccessfully
- #54 redirect_uri in token request

Proposal: Drop these requirements since PKCE is enforced