IETF 116
Yokohama
March 2023

Aaron Parecki
Dick Hardt
Torsten Lodderstedt

OAuth 2.1

https://datatracker.ietf.org/doc/draft-ietf-oauth-v2-1/

draft -08

Changes from draft-07 and IETF 115 Discussion

- Swapped "by a trusted party" with "by an outside party" in client ID definition
- Replaced "verify the identity of the resource owner" with "authenticate"
- Clarified refresh token rotation to match RFC6819
- Added appendix to hold application/x-www-form-urlencoded examples
- Fixed references to entries in appendix
- Incorporated new "Phishing via AS", “Clickjacking”, and “Open Redirection” sections from Security BCP
- Rephrase description of the motivation for client authentication
- Moved "scope" parameter in token request into specific grant types to match OAuth 2.0
- Moved normative requirements out of authorization code security considerations section
- Security considerations clarifications, and removed a duplicate section

Planned Changes for -09

- #64 Finish moving normative language from security considerations inline in the doc
- #136 Finish syncing Security BCP changes into OAuth 2.1
- #97 Expand the differences from OAuth 2.0 to include for which roles each change is a breaking change

Still more open issues to discuss!

https://github.com/aaronpk/oauth-v2-1/issues

Discussion Topics

Define “explicit RO authentication” (#139)

From RFC6749 “Client Impersonation” Security Considerations

```
The authorization server SHOULD enforce explicit resource owner
authentication and provide the resource owner with information about
the client and the requested authorization scope and lifetime. It is up
to the resource owner to review the information in the context of the
current client and to authorize or deny the request.
```

Vittorio: What does this mean in practice?

- Is it a full credential prompt regardless of whether one session already exists?
- A selection between existing sessions, if present?

Aaron: Is this supposed to apply only to public clients?

Repeated authorization requests (#140)

From RFC6749 “Client Impersonation” Security Considerations

```
The authorization server SHOULD NOT process repeated authorization
requests automatically (without active resource owner interaction)
without authenticating the client or relying on other measures to
ensure that the repeated request comes from the original client and
not an impersonator.
```

Vittorio: This is unclear. As it currently reads it seems to prohibit things like getting a new authz code silently via iframe (and prompt=none or equivalent UX suppressing mechanism, please ignore the ITP complications for the sake of argument).

Aaron: Is this supposed to apply to only public clients?

Differences between Mobile vs Desktop apps (#142)

Much of Vittorio's feedback in the native apps section stems from the differences in practice of mobile apps and desktop apps.

- Should these recommendations be scoped to mobile apps now that we have more deployment experience since the Native Apps BCP was written?
- Do we all agree that the native apps recommendations are still valid for desktop apps, even if we know people aren’t always following that advice on desktop apps?

IETF 116
Yokohama
March 2023

Aaron Parecki
David Waite

OAuth 2.0 for Browser-Based Apps

(Best Current Practice)

https://datatracker.ietf.org/doc/draft-ietf-oauth-browser-based-apps/
draft -13

OAuth 2.0 for Browser Based Apps

- Includes recommendations for implementers building browser-based apps using OAuth 2.0
- "Browser-based apps" are defined as applications executing in a browser, aka "SPA" or "single-page apps", and may include a backend component

Changes from -11 to -13

- Added a new section about options for storing tokens in the browser
- Added a section discussing why not to use the Cookie API to store tokens
- Added a section on sender-constrained tokens and a reference to DPoP
- Rephrased the architecture patterns to focus on token acquisition
- Corrected some uses of “DOM” vs “browsing context”
- Consolidated CSRF recommendations from security considerations into main part of the doc
- Updated Implicit flow historic note with info about CORS
- Described limitations of Service Worker storage
- Minor editorial changes

Open Issues

- #22 New pattern:
  BFF proxy storing access tokens in browser as HttpOnly cookies
- Storage isolation (next slide)

Storage Isolation

- There is currently no “best” solution for storing tokens in the browser
- Any browsing-context-accessible storage APIs are vulnerable to XSS attacks
- A Service Worker is isolated from the browsing context
- There is no persistent storage API isolated to Service Workers

Should we ask browsers to add a secure storage mechanism for tokens?