IETF 119
Brisbane
March 2024

Aaron Parecki
David Waite
Philippe De Ryck

OAuth 2.0 for Browser-Based Apps

(Best Current Practice)

https://datatracker.ietf.org/doc/draft-ietf-oauth-browser-based-apps/
draft -17

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

Table of Contents Excerpt

Table of Contents Excerpt

Pattern: Backend for Frontend (BFF)

6.1.4.1. Attack Payloads and Consequences

If the attacker has the ability to execute malicious JavaScript code in the application's execution context, the following payloads become relevant attack scenarios:

- Proxying Requests via the User's Browser (See Section 5.1.4)

Note that this attack scenario results in the following consequences:

- Client Hijacking (See Section 5.2.3)

Pattern: Token-Mediating Backend

6.2.4.1. Attack Payloads and Consequences

If the attacker has the ability to execute malicious JavaScript code in the application's execution context, the following payloads become relevant attack scenarios:

- Single-Execution Token Theft (See Section 5.1.1) for access tokens
- Persistent Token Theft (See Section 5.1.2) for access tokens
- Proxying Requests via the User's Browser (See Section 5.1.4)

Note that this attack scenario results in the following consequences:

- Exploiting Stolen Access Tokens (See Section 5.2.2)
- Client Hijacking (See Section 5.2.3)

Pattern: Pure Browser OAuth Client

6.3.3.1. Attack Payloads and Consequences

If the attacker has the ability to execute malicious JavaScript code in the application's execution context, the following payloads become relevant attack scenarios:

- Single-Execution Token Theft (See Section 5.1.1)
- Persistent Token Theft (See Section 5.1.2)
- Acquisition and Extraction of New Tokens (See Section 5.1.3)
- Proxying Requests via the User's Browser (See Section 5.1.4)

The most dangerous payload is the acquisition and extraction of new tokens. In this attack scenario, the attacker only interacts with the authorization server, which makes the actual implementation details of the OAuth functionality in the JavaScript client irrelevant. Even if the legitimate client application finds a perfectly secure token storage mechanism, the attacker will still be able to obtain tokens from the authorization server.

Note that these attack scenarios result in the following consequences:

- Exploiting Stolen Refresh Tokens (See Section 5.2.1)
- Exploiting Stolen Access Tokens (See Section 5.2.2)
- Client Hijacking (See Section 5.2.3)

Changes from -16 to -17

- Applied editorial changes from Filip Skokan's and Louis Jannett's reviews
- Clarification on application of cookie encryption
- Added a security consideration on the use of postMessage referencing Security BCP (thanks to Louis Jannett)

Next Steps

Last call?

https://www.ietf.org/archive/id/draft-ietf-oauth-browser-based-apps-17.html