

draft-ietf-oauth-security-topics

Status

John Bradley, Andrey Labunets, Torsten Lodderstedt

IETF-100
Nov 15 2017, Singapore

What is it?

- Comprehensive overview on open OAuth security topics
- Systematically captures and discusses these security topics and respective mitigations
- Recommends best current practice and OAuth changes & extensions

Structure

- [1. Introduction](#) [3](#)
- [2. Best Practices](#) [4](#)
 - [2.1. Protecting redirect-based flows](#) [4](#)
 - [2.2. Token Leakage Prevention](#) [5](#)
- [3. Recommended Changes to OAuth](#) [5](#)
- [4. Attacks and Mitigations](#) [5](#)
 - [4.1. Insufficient redirect URI validation](#) [5](#)
 - [4.1.1. Attacks on Authorization Code Grant](#) [6](#)
 - [4.1.2. Attacks on Implicit Grant](#) [7](#)
 - [4.1.3. Proposed Countermeasures](#) [8](#)
 - [4.2. Authorization code leakage via referrer headers](#) [9](#)
 - [4.2.1. Proposed Countermeasures](#) [10](#)
 - [4.3. Attacks in the Browser](#) [10](#)
 - [4.3.1. Code in browser history \(TBD\)](#) [10](#)
 - [4.3.2. Access token in browser history \(TBD\)](#) [10](#)
 - [4.3.3. Javascript Code stealing Access Tokens \(TBD\)](#) [11](#)
 - [4.4. Mix-Up](#) [11](#)
 - [4.5. Code Injection](#) [11](#)
 - [4.5.1. Proposed Countermeasures](#) [13](#)
 - [4.6. XSRF \(TBD\)](#) [15](#)
 - [4.7. Access Token Leakage at the Resource Server](#) [15](#)
 - [4.7.1. Access Token Phishing by Counterfeit Resource Server](#) 15
 - [4.7.1.1. Metadata](#) [16](#)
 - [4.7.1.2. Sender Constrained Access Tokens](#) [17](#)
 - [4.7.1.3. Audience Restricted Access Tokens](#) [19](#)
 - [4.7.2. Compromised Resource Server](#) [20](#)
 - [4.8. Refresh Token Leakage \(TBD\)](#) [21](#)
 - [4.9. Open Redirection \(TBD\)](#) [21](#)
 - [4.10. TLS Terminating Reverse Proxies](#) [22](#)

} Recommendations

} Threat Analysis and Discussion of Counter Measures

Recommended Best Practices

- Exact redirect URI matching at AS (token leakage, mix-up)
- Avoid any redirects or forwards, which can be parameterized by URI query parameters (open redirection, token/code leakage)
- One-time use tokens carried in the STATE parameter for XSRF prevention
- AS-specific redirect URIs (mix-up)
- Clients shall use PKCE in order to detect code injection
- Authorization servers shall use TLS-based methods for sender constraint access tokens
- Use end-to-end TLS whenever possible

Recommended Changes to OAuth

Remove requirement to check actual redirect URI at token endpoint (RFC 6749, Section 4.3.1)

“ensure that the "redirect_uri" parameter is present if the "redirect_uri" parameter was included in the initial authorization request ... and if included ensure that their values are identical.”

- Objective: prevent client impersonation/code injection
- Challenges:
 - seems to complicated to implement properly and requires transaction specific state
 - Some implementations short cut be just pattern matching against client policy
- protection goal is achieved more effective by utilizing PKCE as recommended

Status

- Published revisions -03 & -04
- Added text on token leakage at the resource server based on discussions in Prague
 - Additional Metadata
 - Audience Restriction
 - Sender Constraint Access Tokens
- Added text on threats associated with use of TLS Terminating Reverse Proxies
- Based on feedback in Prague, added recommendation to use sender constraint access tokens as preferred countermeasure against token leakage
- Restructured BCP

Please give us feedback!