draft-ietf-oauth-security-topics-12

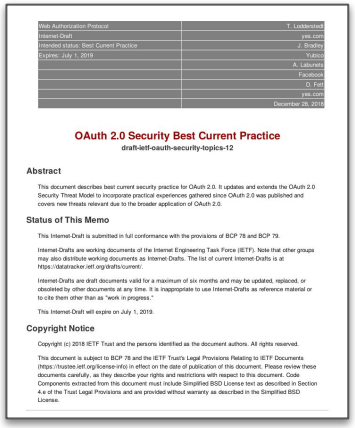
OAuth 2.0 Security Best Current Practice

Torsten Lodderstedt, John Bradley,
Andrey Labunets, Daniel Fett

IETF-104
Mar 25 2019, Prague

OAuth 2.0 Security Best Current Practice

- Refines and enhances security guidance for OAuth 2.0 implementers
- Updates, but does not replace:
  - OAuth 2.0 Threat Model and Security Considerations (RFC 6819)
  - OAuth 2.0 Security Considerations (RFC 6749 & 6750)



Web Authorization Protocol	T. Lodderstedt
Internet-Draft	yes.com
Intended status: Best Current Practice	J. Bradley
Expires: July 1, 2019	Yubico
	A. Labunets
	Facebook
	D. Fett
	yes.com
	December 28, 2018

OAuth 2.0 Security Best Current Practice

draft-ietf-oauth-security-topics-12

Abstract

This document describes best current security practice for OAuth 2.0. It updates and extends the OAuth 2.0 Security Threat Model to incorporate practical experiences gathered since OAuth 2.0 was published and covers new threats relevant due to the broader application of OAuth 2.0.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 1, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

- Updated, more comprehensive Threat Model
- Description of Attacks and Mitigations
- Simple and actionable recommendations

Changes Since IETF-103 (-09 … -12)

- WG Consensus to discourage use of implicit and the like

“… clients SHOULD NOT use the implicit grant (response type "token") or any other response type issuing access tokens in the authorization response, such as "token id_token" and "code token id_token", unless the issued access tokens are sender-constrained and access token injection in the authorization response is prevented.”

- Recommendation: use code + PKCE for SPAs
- Adoption has started
  - OpenID FAPI WG removed “code token id_token” from Part 2 spec
  - oidc-client-js library added support for code+PKCE

Changes since IETF-103 (contd.)

- Added text on refresh tokens including replay detection based on refresh token rotation
- Added refined attack model incorporating more dynamic setups
- Migrated draft to markdown

But we are not done yet...

Proposed Changes

- Discourage use of Resource Owner Password Credentials Grant Type
- Recommend client authn methods based on public key crypto
  - mTLS
  - private_key_jwt
- Use PKCE for CSRF prevention instead of state parameter
  - PKCE is mandatory now and can fulfill this additional task
  - Simplifies recommendations and makes state available again for original purpose (carry client transaction data)
- Recommend PKCE mode S256 over plain
  - Attacker cannot utilize leaked code_challenge to complete transaction

Further Challenges (FYI)

- OAuth in SPAs
  - SPA BCP Draft (https://tools.ietf.org/html/draft-ietf-oauth-browser-based-apps-00)
  - Mechanisms for sender-constrained tokens using application level signing needed (TBD)
- Secure Transaction Authorization/Rich Authorization Requests
  - Lodging Intent Pattern (OIDF FAPI)
  - https://bitbucket.org/openid/fapi/src/master/Financial_API_Lodging_Intent.md
- JWT protected authorization responses
  - JWT Secured Authorization Response Mode for OAuth 2.0 (JARM) (OIDF FAPI)
  - https://bitbucket.org/openid/fapi/src/master/Financial_API_JWT_Secured_Authorization_Response_Mode.md

Q & A