IETF 125
Shenzhen
March 2026

Updates to OAuth 2.0 Security Best Current Practice

Tim Würtele[1] Pedram Hosseyni[1] Kaixuan Luo[2] Adonis Fung[3]

[1]University of Stuttgart, Germany

[2]The Chinese University of Hong Kong, Hong Kong SAR, China

[3]Samsung Research America, USA

https://datatracker.ietf.org/doc/draft-ietf-oauth-security-topics-update/

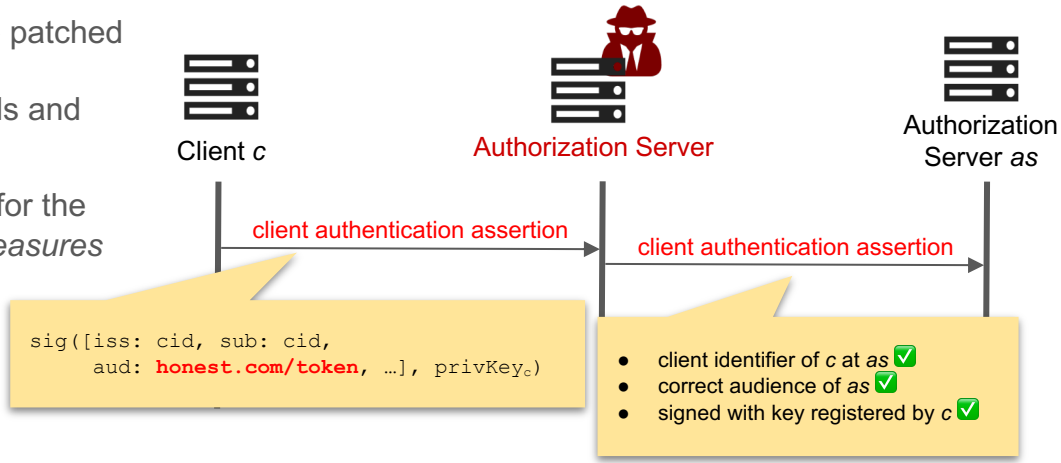
draft -01

Background / History

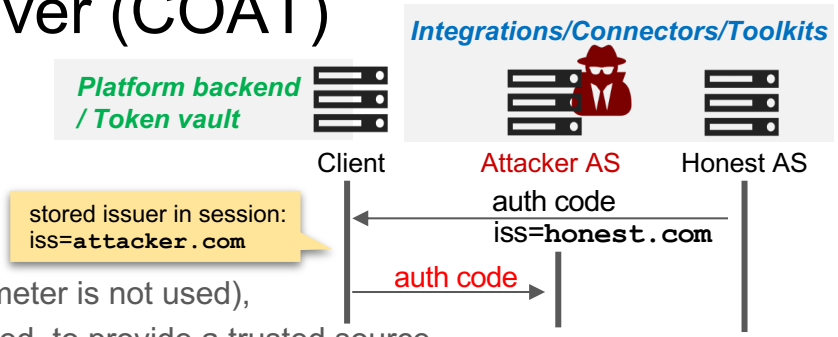
- New attacks since RFC9700 is published
- Jan 2025: Interim meeting on Audience Injection Attack **[Section 2.1]**
  - https://datatracker.ietf.org/meeting/interim-2025-oauth-04/session/oauth
- Sep 2025: Interim meeting on security issues in "Account Linking" **[Section 2.2 & 2.3]**
  - https://datatracker.ietf.org/meeting/interim-2025-oauth-08/session/oauth
- Also discussed at OSW, IETF 122 & 123
- Oct 2025: WG-Adoption Call
- Expectation (summarized based on previous WG feedback):
  - Not meant to replace RFC9700
  - Can collect multiple Security Best Current Practice RFCs under BCP 240
  - As best "current" practice, need not hold until all new attacks come out; keep a small focused doc to publish

Audience Injection Attacks [Section 2.1]

- RFC7523bis:
  - WG state: Submitted to IESG for Publication
  - Only talks about how RFC7523 shall be patched
  - Does not mention how the attack works
  - Does not specify other affected protocols and alternative mitigations
- This document: a security-centric perspective for the problem, including both the *attack* and *countermeasures*
  - Detailed description of the attack
    - Attack setting and requirements
    - Description of how the attacker obtains a valid client authentication assertion
    - List of example endpoints at which the attacker can obtain such assertions
  - Proposed countermeasures
    - Single audience value
    - Issuer identifier
    - Exact URI of the target endpoint



Cross-toolkit OAuth Account Takeover (COAT)



- Existing security guidance on Mix-Up attacks (Section 4.4 of RFC9700) relies on ***issuer identifier*** to uniquely identify an AS and avoid confusion
- In real-world deployments, the issuer identifier:
  - may not be ***available*** (if RFC9207 – `iss` response parameter is not used),
  - may not be ***trusted*** (if RFC8414 – AS Metadata is not used, to provide a trusted source for stored `iss` value, but instead w/ manually-configured AS endpoints),
  - may not be ***unique*** (if multiple AS instances, configured separately at the client, use the same AS)

  ⇒ These hinder real-world vendors from applying the issuer-based defenses (**none** of the 20+ OAuth client vendors we investigated did)
- **Real-world attacks are prevalent:** attack precondition of "a client interacts with multiple AS, one of which is malicious" can be easily satisfied in certain open ecosystems:

e.g.,
  - **Cloud Platforms** for Application Integration: open marketplace of SaaS integrations (aka connectors)
  - **Token Vaults** in Agentic AI: each tenant can equip its applications/AI agents with custom toolkits

  ⇒ Malicious AS can be introduced via such integrations / connectors / toolkits, causing unauthorized access

  ⇒ **20+ Vulnerable Vendors:** Microsoft Power Automate, Amazon Alexa, Google Home, Microsoft Copilot Studio, Amazon Bedrock AgentCore, Arcade.dev, etc.

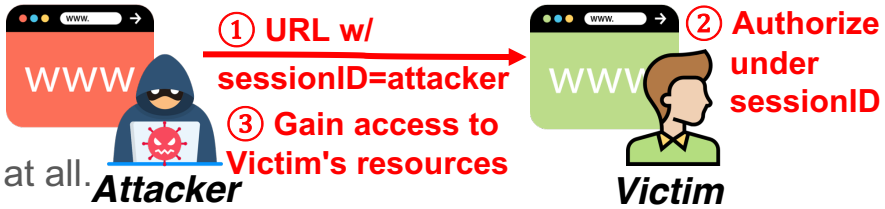
Cross-toolkit OAuth Account Takeover (COAT) [Section 2.2]

Section 2.2 is a self-contained section, which:

- Model the aforementioned **common deployment scenarios** (as non-normative examples) for community awareness
- Provide generalized **attack descriptions** under the name "Cross-toolkit OAuth Account Takeover (COAT)"
- Propose a handy **countermeasure** an OAuth Client can readily apply to protect **existing** RFC6749-compliant OAuth 2.0 deployments:
  - A client-assigned identifier (***available*** and ***trusted***) for each integration / connector / toolkit, that ***unique***ly identifies each AS instance, rather than the issuer identifier
  - Embedded in `redirect_uri`
  - Matching enforced at the client in OAuth flows
- Delineate in what cases the issuer identifier (existing mix-up countermeasures) is preferred
  - Standards-compliance of RFC9207 (and RFC8414) from both clients and ASs are met
- Corrected some misconceptions in Section 4.4 of RFC9700
  - *e.g.,* "the client uses the same redirection URI for each AS" should not be an attack precondition

Cross-user OAuth Session Fixation

- Session Fixation is a known threat [1] in OpenID4VP (response_mode=direct_post), Cross-Device Flows, as well as OAuth 1.0 (Request Token approval flow).
  - Attacker tricks a victim into authorizing an OAuth flow initiated by the attacker, which fixates an attacker-owned auth session in victim's user-agent
  - After victim's authorization, attacker gains access to victim's protected resources
- ***In Theory***, OAuth 2.0 authorization code grant in RFC6749 mitigates Session Fixation by not inventing an auth session in the first place, only leaving `state` parameter to maintain application state, which should also be securely bound to the user-agent session.
- Therefore,
  - RFC9700 does **not** discuss Session Fixation at all.
  - It only mentions (Login) CSRF, where `state` serves as a CSRF token.



[1] https://danielfett.de/2025/03/10/cross-device-session-fixation/

Cross-user OAuth Session Fixation

- ***In Reality***, however, many vendors introduce standalone auth sessions due to the need to carry application state, especially to encode information about the authorization context, such as the recipient user of the token at the client.
  These auth sessions are often introduced either in `state`, or as separate URL query parameters.
- **Real-world attacks are prevalent:** Clients often fail to bind auth sessions to the user-agent session, causing Session Fixation.

  Not merely careless implementation flaw, but ***unable*** to do so, as the (1) *OAuth responsibilities* and (2) *session management* are decoupled and handled by separate components of the client application:

e.g.,
  - **Cloud Platforms** for Application Integration: separate services isolated under different origins, and/or accessed from different user-agents.
  - **Token Vaults** in Agentic AI: each tenant manages its own users for its applications/AI agents, with OAuth responsibilities outsourced to the Token Vault.

⇒ Session Fixation feasible by sharing the request URL that encodes the auth session, either the authorization request URL, or the request URL preceding it.

⇒ **40+ Vulnerable Vendors:** Amazon Alexa, IFTTT, Microsoft Copilot, n8n, Perplexity, Manus, Amazon Bedrock AgentCore, Arcade.dev, Composio, etc.

Cross-user OAuth Session Fixation [Section 2.3]

In Section 2.3, we:

- Model **common deployment scenarios**
- Provide **attack descriptions**, including both cases where auth sessions can be introduced
- Propose **countermeasures**:
  - (If Cross-Origin / UA), Return to the ***web origin*** or ***user-agent*** where the client's original session is available
  - Validate the binding of *auth session* to *existing (user) session* at the client
  - **Note:** Currently, the normative language does **not** specify concrete protocol flows, but rather high-level defense descriptions, because coordination between different components of the client application is often proprietary, and thus considered out of scope of this BCP document. (#38)
  - (unless we want to further profile the flow, just like the BFF pattern in Browser-based Apps)
- Delineate attack's relationship to CSRF, and PKCE's inadequacy to resolve the problem alone

Document Updates

-00

- WG adoption, no changes from previous individual draft

-01

- Clarify that shared redirection URI is not a precondition of COAT (#43)
- Clarify that COAT countermeasure uniquely identifies each configured AS instance (#41)
- Clarify Session Fixation countermeasures and relationship to CSRF and PKCE (#48)
- Use terminology that is less ambiguous (e.g., tool -> toolkit) and better aligned with standard OAuth language (e.g., platform -> application) (#44)
- Editorial clarifications (#44) and fixes (#42, #46), and reference updates (#45)

Path Forward

- Comments on the draft content are greatly appreciated
  - https://datatracker.ietf.org/doc/draft-ietf-oauth-security-topics-update/
  - Further Discussion: https://github.com/SECtim/draft-wuertele-oauth-security-topics-update/issues
- Next Steps?

  Expectation (summarized based on previous WG feedback):
  - Not meant to replace RFC9700
  - Can collect multiple Security Best Current Practice RFCs under BCP 240
  - As best "current" practice, need not hold until all new attacks come out; keep a small focused doc to publish
- We plan to collect feedback on the draft and incorporate it / resolve all issues before IETF 126, where we'd like to discuss any remaining issues and, depending on the feedback, see whether the document will be ready for WG last call :)