OAuth Mix-Up

John Bradley

Documents

- Draft https://tools.ietf.org/html/draft-ietf-oauth-mix-up-mitigation-00

- Mladenov, V., Mainka, C., and J. Schwenk, "On the security of modern Single Sign-On Protocols: Second-Order Vulnerabilities in OpenID Connect", arXiv 1508.04324v2, January 2016, <http://arxiv.org/abs/1508.04324v2/>.

- Fett, D., Kuesters, R., and G. Schmitz, "A Comprehensive Formal Security Analysis of OAuth 2.0”, arXiv 1601.01229v2, January 2016, <http://arxiv.org/abs/1601.01229v2/>.

Things for Clients to be Mixed-Up about

- There are multiple variations of the attack, resulting in the client being confused about one or more of:
  - Dynamic registration endpoint
  - Authorization endpoint
  - Token Endpoint
  - Resource Endpoint

Attackers goals

- Leverage the users trust in the client and AS being attacked.
- Leverage an existing sticky grant that the user has in the AS for the client to have a token issued without user interaction.
- Get access to the API directly
- Get access to the API indirectly via binding a new account to the API via the client.

Authorization endpoint MiM Cause

- The client “remembers” who it made the request to
  - This can be stored in state or in a cookie
- The client assumes that the response is coming from the AS the request was made to, and has no way to detect a modification of the request or response.
- An attacker can use this to MiM the Authorization request (typically to modify client_id)

Token endpoint and RS endpoint MiM

- This is caused by malicious configuration information

Preconditions

- Typically the client needs to be vulnerable to having a 3rd party trigger an authorization.
    - improper xsrf protection on input forms or pages without TLS can be used by attackers to start an attack.

- Clients need to have more than one client_id (get authorizations from more than one AS)

Dynamic registration

- A client doing dynamic registration is easier to attack because the attacker can potentially trick it into registering at a bad AS

- The same thing can be done via manual client registration or compromising a existing AS.

Discovery

- Potentially makes it easier to automate an attack by giving a client bad endpoint information.
- Not required for an attack.
    - Bad endpoints can be manually configured by developers.

Client identification

- Some variations of this and other attacks take advantage of the AS having quite weak ways of identifying the client to the user in the Consent dialog.

- This may be a more general problem than mix-up

Possible Mitigations for Authorization and token endpoints

- Identifying the AS and the client_id in the authorization response

- Integrity protecting Authorization Requests and or responses

- Enforce one client_id per redirect_uri/client

Possible Mitigations for RS

- Audience restrictions on bearer AT
    - https://tools.ietf.org/html/draft-campbell-oauth-resource-indicators-01
- PoP AT
- Out of band validation of RS
    - https://tools.ietf.org/html/draft-hunt-oauth-bound-config-00