Token Binding for OAuth and OpenID Connect and Implications for the Token Binding Protocol

Mike Jones

IETF 96, Berlin

July 2016

Goals

- Specify means of using Token Binding with OAuth and OpenID Connect
    - Token Binding of access tokens, refresh tokens, ID tokens
    - (Brian Campbell will separately describe Token Binding of OAuth authorization codes)
- Do this before Token Binding finishes to
    - enable end-to-end testing
    - identify any gaps in Token Binding for these use cases
        - One possible gap identified will be discussed shortly

Token Bound Refresh Tokens

- Simplest of the cases
  - Defined in draft-jones-oauth-token-binding
- Two parties using a TLS connection:
  - Client and Authorization Server (AS)
- AS adds token binding info to refresh tokens sent to client
- AS checks token binding info when refresh tokens sent by client to AS
- Transparent to client!

Token Bound ID Tokens

- Next simplest case
  - Defined in openid-connect-token-bound-authentication-1_0
- Three parties using two TLS connections:
  1. User Agent (UA) and Relying Party (RP)
  2. User Agent (UA) and Identity Provider (IdP)
- RP sends request to IdP using 302 redirect
  - HTTPS Token Binding protocol sends UA/IdP Token Binding to IdP as referred token binding
- IdP puts Referred TB info in ID Token
- RP validates TB info in ID Token

Representation in ID Token

- SHA-256 hash of Token Binding ID added in “cnf” (confirmation) claim value in ID Token
- No need to include full Token Binding ID, since carried in referred token binding information
- Hashing TBID makes even RSA TBIDs small enough to be reasonable to include in ID Token

- Open Issue: Enabling crypto agility for hash function

Token Bound Access Tokens

- More complicated case that raises an open issue
  - Defined in [draft-jones-oauth-token-binding]
- Three parties using two (or three) TLS connections:
  1. Client and Resource Server (RS)
  2. Client's User Agent and Authorization Server (AS) AuthZ Endpoint
  3. Client and Authorization Server (AS) Token Endpoint
- Client learns token binding info for conn. 1
- Client sends request to AS on conn. 2 with conn. 1 TB info
  - For now, sent as an explicit request parameter
- AS puts conn. 1 TB info in access token delivered to client over conn. 2 or conn. 3 (depending upon OAuth response_type)
- Client uses access token at RS over conn. 1
- RS validates TB info in access token

Issue for Token Bound Access Tokens

- **Issue:** Unlike ID Token case, in which referred token binding sent via 302 redirect, in this case, client doesn't use redirections for cross-domain communication
  - Cross-domain communication by explicit communication on different channels
  - Referred token binding not sent
  - Instead, token binding info sent via explicit request parameter
- **Problem:** Two channels not cryptographically bound together when using explicit parameter method
- **Proposed Solution:** Require Token Binding implementations to provide APIs for clients to explicitly provide TB info to be sent as referred token binding
  - Gives applications the same functionality as used by 302 redirect
- *Also applies to OAuth Token Exchange, other protocols*

Solution Applicability

- Enabling explicit cross-origin Token Bound communication would be used for OAuth access tokens
- Many other applications communicate across multiple channels
- This functionality should be widely useful & used
- Without it, many applications couldn’t secure cross-origin communication with Token Binding

Issue Discussion

- Are people in favor of the solution?
- Do people see problems with it?
- What are the next steps?

Possible Future Work

- Token Binding for JWT Client Authentication [RFC 7523]
- Token Binding for JWT Authorization Grants [RFC 7523]
- Token Binding for OAuth Token Exchange [draft-ietf-oauth-token-exchange]
- Token Bound Client IDs issued by OAuth Dynamic Client Registration [RFC 7591]
- *Anything else we should include?*