Torsten Lodderstedt (presenter & co-author)
Brian Campbell (editor & workation photographer)
John Bradley (co-author)
Nat Sakimura (co-author)

OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens

draft-ietf-oauth-mtls

IETF 104
Prague
March 2019

OAuth MTLS
Context & Overview

- Why?
  - Enhanced security for OAuth 2.0 based on TLS client certificates
    - Draft is already being used by OpenBanking/PSD2esque regulatory regimes and other SDOs
- What?
  - Asymmetric key based client authentication to the AS using mutual TLS
    - Two methods:
      - PKI based mode using subject distinguished name or subject alternative name
      - Self-signed certificate mode
  - Mutual TLS certificate bound access tokens for proof-of-possession protected resources access
    - “x5t#S256”: X.509 Certificate SHA-256 Thumbprint Confirmation Method for JWT and Introspection
  - Mutual TLS certificate bound refresh tokens for public clients

Since we last met in Bangkok

Draft -13 published addressing:

- **AD review**
- **A few sometimes ‘spirited’ mailing list discussions**

Changes in -13 since IETF 103 Bangkok
(slide 1 of 2)

- Intro has a new abstract protocol flow diagram to serve as an overview of OAuth in general and baseline to describe the various ways in which the MTLS mechanisms are intended to be used
- Added new client metadata values to allow for the use of a SAN in the PKI MTLS client authentication method
- Added an "mtls_endpoint_aliases" AS metadata parameter that is a JSON object containing alternative authorization server endpoints, which a client intending to do mutual TLS will use in preference to the conventional endpoints
- Move the explanation about "cnf" introspection registration into the IANA Considerations

Changes in -13 since IETF 103 Bangkok
(slide 2 of 2)

- Added description of binding refresh tokens for public clients in 'Public Clients and Certificate Bound Tokens' top-level section (moved and renamed from implementation considerations)
- New privacy considerations attempting to discuss implications of the client cert being sent in the clear in TLS 1.2
- New security considerations section about the certificate thumbprint binding (and moved the hash algorithm agility recommendation there)
- Editorial updates and improvements
  - Reword/restructure main PKI method section
  - A bit less German influence
  - Reword/restructure the Self-Signed method section
  - Reword implementation considerations somewhat

Looking ahead to IETF 105 Montreal

- Add a little explanation about how, with tls_client_auth & self_signed_tls_client_auth, refresh tokens are certificate-bound indirectly via the client authentication?
- Hopefully done!?

Q&A

Metadata for Mutual TLS Endpoint Aliases

```
{
  "issuer": "https://server.example.com",
  "authorization_endpoint": "https://server.example.com/authz",
  "token_endpoint": "https://server.example.com/token",
  "introspection_endpoint": "https://server.example.com/introspect",
  "revocation_endpoint": "https://server.example.com/revo",
  "jwks_uri": "https://server.example.com/jwks",
  "response_types_supported": ["code"],
  "response_modes_supported": ["fragment","query","form_post"],
  "grant_types_supported": ["authorization_code", "refresh_token"],
  "token_endpoint_auth_methods_supported": ["tls_client_auth","client_secret_basic","none"],
  "tls_client_certificate_bound_access_tokens": true
  "mtls_endpoint_aliases": {
   "token_endpoint": "https://mtls.example.com/token",
   "revocation_endpoint": "https://mtls.example.com/revo",
   "introspection_endpoint": "https://mtls.example.com/introspect"
  }
 }
```