Authority Tokens for ACME

IETF 102

ACME WG

Jon - Montreal - Jul 2018

Authority Token Challenge

- Identified a generic need for authorities to provide tokens to a CA to respond to challenges
  - Surely any number of namespaces have authorities who could generate tokens
    - Inspired by the STIR case, but this could work for domains even
  - Requires the ACME server has some trust relationship with the authority
- draft-ietf-acme-authority-token-00
  - Framework for tokens that allow authorities trusted by the CA to attest client ownership of names
    - CA can then issue certs via ACME for particular names
  - Need some sort of typing mechanism for tokens, and a means to contact authorities

Example challenge

```
"challenges": [
        {
          "type": "tkauth-01",
          "tkauth-type": “ATC",
          "token-authority": "https://authority.example.org/authz",
          "url": "https://boulder.example.com/authz/asdf/0"
          "token": "IlirfxKKXAsHtmzK29Pj8A" }
        ]
```

- The tkauth-type is governed by a registry
  - Specifies the syntax of the token
    - Today we only specify one initial registration, for JWT (do we need more?)
  - It is the identifier type in the challenge that tells you what you are asking the authority to attest
- The token-authority contains an optional URL
  - A hint for where clients can get a token
  - Not mandatory to follow, clients may already know where to get tokens from some out-of-band source

The “ATC” tkauth-type

- “ATC” tkauth-type based on JWT
  - Used by the TNAuthlist document
- Example ACME response with a JWT
  - The JWT itself is the “ATC” payload in **bold**

```
{ "protected": base64url({
  "alg": "ES256",
  "kid": "https://boulder.example.com/acme/reg/asdf",
  "nonce": "Q_s3MWoqT05TrdkM2MTDcw",
  "url": "https://boulder.example.com/acme/authz/asdf/0" }),
  "payload": base64url({ "ATC": "evaGxfADs...62jcerQ" }),
  "signature": "5wUrDI3eAaV4wl2Rfj3aC0Pp--XB3t4YYuNgacv_D3U" }
```

Fingerprint v. Nonce

- We discussed this issue last time
- Now there is a “binding” of the Authority Token JWT to the ACME
  - Assumes fingerprint of the credentials of the ACME account is the default choice
  - Other profiles might want to use nonce
  - Might want other bindings, specific to resources?
- This has some design implications
  - Fingerprint works per account
  - Nonce works per challenge instead
    - You need a new ATC token for each challenge
      - Could be a lot of work for short-lived certs
- Any further thoughts?

Token Authority interface

- We want to have at least one mechanism for requesting a token from a Token Authority
  - Right now there's a [TBD] for this
  - Not mandatory-to-use, but a baseline
- ReST API seems simplest
  - ATIS has done some work on this, will copy it
  - Based on the principles that you ask the Token Authority to sign for the ASN.1 object we expect will populate the cert
- The next presentation will talk more about that…

NOW FOR CHRIS