JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens

Vittorio Bertocci

IETF 104

March the 25th

Context

- Lots of real-world authorization servers issue access tokens as JWTs
  - Same functional layout, but proprietary claim types, validation requirements..
- Many specs postulate the existence of JWT access tokens but are light on actionable guidance
  - E.g. audience restrictions requirements
- Defining a JWT profile for access tokens would
  - Improve cross-vendor interoperability (SKDs, tools, etc)
  - Prevent issues in today's proprietary implementations
    - E.g. using access tokens in lieu of id_tokens
  - Provide a layout other specs can refer to

The work so far

- Gathered examples of JWT access tokens from multiple vendors
- Distilled a proposal reflecting the requirements implied by those examples of real-world usage
- Discussed the proposal extensively at OSW2019 last week
  - Deck: https://sec.uni-stuttgart.de/_media/events/osw2019/slides/bertocci_-_a_jwt_profile_for_ats.pptx
- Captured the feedback in one draft & submitted
  - Draft: https://tools.ietf.org/html/draft-bertocci-oauth-access-token-jwt-00

Profile Overview

JWT Access token layout (1/2)

claim name		etymology	function
iss	REQUIRED	OpenID.Core	validation
exp	REQUIRED	OpenID.Core	
aud	REQUIRED	OpenID.Core, resource indicators	
iat	OPTIONAL	OpenID.Core	
auth_time	OPTIONAL	OpenID.Core	
sub	REQUIRED	OpenID.Core	identity
<identity claims>	OPTIONAL	OpenID.Core	
scope	when scope is present in the request, REQUIRED	token exchange	authorization
groups, roles, entitlements	OPTIONAL	SCIM Core 7643	
client_id	REQUIRED	token exchange	context
jti	OPTIONAL	JWT 7519	
acr, amr	OPTIONAL	OpenID.Core	

JWT Access token layout (2/2)

- JWT access tokens MUST have a typ header value of **at+jwt**
- This is to prevent resource servers from accepting access tokens as id_tokens

JWT Access Token Layout - Minimal

- Smallest possible JWT AT when scopes are requested


```
{"typ":"at+jwt","alg":"RS256","kid":"RjEwOwOA"}
{
    "iss": "https://authorization-server.example.com/",
    "sub": " 5ba552d67",
    "aud":   "https://rs.example.com/inbox",
    "exp": 1544645174,
    "client_id": "s6BhdRkqt3_",
    "scope": "openid profile reademail"
  }
```

Requesting JWT Access Tokens

- Any existing grant returning an access token can return a JWT access token
- If a request contains **resource**, its value must be reflected in **aud**
  - No multi-value **resource** admitted in reqs for JWT access tokens (scope confusion)
- Without **resource** in the req, the authorization server either
  - Infers the resource indicator from **scope** and assign it to **aud**
    - All scope strings must refer to the same resource
  - Or assigns a default value
- If a request contains **scope**, the resulting JWT access token must feature a **scope** claim
- Whether to include identity claims, non-delegation claims or custom claims is an agreement between authorization server and resource server
  - The client has no say on the matter

Validating JWT Access Tokens

- When possible, advertise keys and **iss** via RFC8414 or OIDC discovery
- Check
  - typ==at+jws
  - iss
  - aud
  - signature
  - exp
  - [optional] auth_time
  - [discussion] scope?

Security Considerations

- Typ=at+jwt prevents id_token confusion
- Enforcing single audience prevents scope-resource confusion

Privacy considerations

- Encrypt when clients shouldn’t see token content
- If embedding extra identity claims, ensure that the resource server can lawfully see that info

Next steps

- WG adoption?
- Overall validation
- Open issues
  - Should we say something about the RS responsibility to act on scopes?
  - What errors codes to use?
  - Can we/should we specify whether the client was confidential?
  - Can we/should we specify that the use comes from a federated IdP?

https://tools.ietf.org/html/draft-bertocci-oauth-access-token-jwt-00