One-Time Confirmation Tokens

Dmitry Telegin, Backbase UK

IETF 121 Dublin

One-Time Tokens…

…And Where To Find Them

One-time objects in OAuth:

- authorization codes
- DPoP proofs
- **access(-ish) tokens**

Confirmation Tokens

Examples

- Payments
- Disclosing sensitive data (e.g. card details)
- Generating statements
- Adding trusted recipients
- Setting Pay PIN
- Updating profile information
- ...

Confirmation Tokens

…vs. Access Tokens

	Access Tokens	Confirmation Tokens
Lifetime	Short-lived (e.g. minutes)	Very short lived (e.g. 30 sec)
Times used	Multiple-time	One-time
Refresh	Yes	No
Scope	API	Operation
Session	Yes	No
Use	Standalone	Complementary (w/access token)

Confirmation Tokens

…vs. Step Up

Step Up:

- Access token is replaced
- “acr” upgraded

Confirmation:

- Access token is retained / refreshed
- “scope”, “authorization_data” upgraded
  - “acr” retained or upgraded

Confirmation Tokens

What People Do

- start separate auth session
- pass access token
- request custom scopes
- pass as custom header
- revoke “confirmation” token after first use

```
GET /as/auth?client_id=...&scope="bb:confirm:XXXXXX bb:deny"
```

```
GET /rs/foo

Authorization: Bearer XXX
X-BB-Confirm: YYY
```

Confirmation Tokens

Resource Server

```
> POST /rs/payments
> Authorization: Bearer XXX
> ...

< HTTP/1.1 403 Forbidden
< WWW-Authenticate: Bearer, error="confirmation_required", scope="...",
<     authorization_details="...", scheme="DPoP algs=\"ES256 PS256\""
```

Confirmation Tokens

Authorization Server

```
GET /as/auth?client_id="foo"&scope="openid ..."&...&confirmation=true
```

```
> POST /as/token
> grant_type=confirmation_code&
> client_id=s6BhdRkqt&
> code=SplxlOBeZQQYbYS6WxSbIA&
> refresh_token=tGzv3JOkF0XG5Qx2TlKWIA

< {
<  "access_token": "XXXX",
<  "token_type": "Bearer",
<  "refresh_token": "XXXX",
<  "id_token": "XXXX",
<  "conf_token": "XXXX",
<  ...
< }
```

Confirmation Tokens

What's inside

```
{
  “kid”: “XXX”,
  “typ”: “ct+jwt”
}
.
{
  “jti: “YYY”
  "sub":"someone@example.com",
  "iss":"https://server.example.com",
  "nbf":1562262611,
  "exp":1562266216,
  “use”: 1,
  “at_hash”: “XXXXXXXXXX”,
  “scope”: “confirm:ZZZ”,
  "cnf":
  {
    "jkt":"0ZcOCORZNYy-DWpqq30jZyJGHTN0d2HglBV3uiguA4I"
  }
}
.
<SIG>
```

Confirmation Tokens

Client Request

```
> POST /rs/payments

> Authorization: DPoP XXX
> Confirmation: DPoP YYY
> ...

< HTTP/1.1 201 Created
```

Confirmation Tokens

Introspection

```
> POST /as/introspect HTTP/1.1
> Host: server.example.com
> Accept: application/json
> Content-Type: application/x-www-form-urlencoded
> Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

> token=mF_9.B5f-4.1JqM&token_type_hint=conf_token

< HTTP/1.1 200 OK
< Content-Type: application/json

< {
<   "active": true,
<   "client_id": "l238j323ds-23ij4",
<   "scope": "openid confirm:XXX",
<   "sub": "Z5O3upPC88QrAjx00dis",
<   "exp": 1419356238,
<   "iat": 1419350238,
<   “use”: 1,
<   “uses_left”: 0
< }
```

Confirmation Tokens

Privacy considerations

- AS becoming “Big Brother”?
- not reinventing 3DSecure
- first-party ~~apps~~ everything (AS + RS + client)
  - integration with OAuth FiPA

Appendix: Action Tokens

Use Cases

- Magic link login
- Password reset
- Confirm email
- Invite to an organization

Action Tokens

…vs. Confirmation Tokens

	Action Tokens	Confirmation Tokens
Lifetime	Long-lived (e.g. hours)	Very short lived (e.g. 30 sec)
Times used	One-time	One-time
Refresh	No	No
Scope	Operation (AS)	Operation (RS)
Use	Standalone	Complementary (w/access token)
Subject	Yes/No	Yes

Action Tokens

What's inside

```
{
  “kid”: “XXX”,
  “typ”: “act+jwt”
}
.
{
  “jti: “YYY”
  "sub":"someone@example.com",
  "iss":"https://server.example.com",
  "nbf":1562262611,
  "exp":1562266216,
  “scope”: “as:confirm-email”
  “use”: 1
}
.
<SIG>
```