OAuth 2.0 Demonstration of Proof-of-Possession at the Application Layer (DPoP)

Brian Campbell (presenter, co-author, workation photographer)
Torsten Lodderstedt (co-author)
Daniel Fett (co-author)

John Bradley (co-author of sorts)
Michael Jones (co-author)
David Waite (co-author)

IETF 106
Singapore
Nov 2019
DPoP is a draft proposal for a new[ish], simple and concise approach to proof-of-possession for OAuth access and refresh tokens using application-level constructs and leveraging existing library support.

-00 was published during IETF 105 in Prague thereby justifying the use of this photo.
Prior proof-of-possession efforts in OAuth:
The road to now is littered with [to varying degrees] failures

- “OAuth 1.0a” - RFC 5849
- “OAuth 2.0 Message Authentication Code (MAC) Tokens” - draft-ietf-oauth-v2-http-mac
- “Proof-of-Possession Key Semantics for JSON Web Tokens” – RFC 7800
- “OAuth 2.0 Proof-of-Possession (PoP) Security Architecture” - draft-ietf-oauth-pop-architecture
- “OAuth 2.0 Proof-of-Possession: Authorization Server to Client Key Distribution” - draft-ietf-oauth-pop-key-distribution
- “A Method for Signing HTTP Requests for Oauth” – draft-ietf-oauth-signed-http-request
- “OAuth 2.0 Token Binding” - draft-ietf-oauth-token-binding
- “OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens” - draft-ietf-oauth-mtls
Motivations for this new effort

- Be better than bearer (be best...?)
- OAuth 2.0 Security BCP recommends use of sender-constrained tokens (somewhat aspirational)
  - To prevent token replay at a different endpoint/resource (among other benefits)
- Yet OAuth lacks suitable and widely-applicable PoP mechanism
- Especially true for Single Page Applications (SPA)
  - MTLS for OAuth 2.0 would have major UX issues with SPAs
  - Status of Token Binding is uncertain
- Proof-of-possession bound refresh tokens for public clients
Basic DPoP flow in ASCII

Client

--(A)-- Token Request ------------------------>
(DPoP Proof)

<- (B) -- DPoP-bound Access Token -------->
(token_type=DPoP)
PoP Refresh Token for public clients

--(C)-- DPoP-bound Access Token ---------->
(DPoP Proof)

<- (D) -- Protected Resource ---------------

Authorization Server

Resource Server
Anatomy of a DPoP Proof JWT

{  
  "typ": "dpop+jwt",
  "alg": "ES256",
  "jwk": {
    "kty": "EC", "crv": "P-256",
    "x": "l8tFrhx-34tV3hRlCRDY9zCkDlpBhF42QfWVAWBFs",
    "y": "9VE4jfOk_o64zbTT1cuNJajHmt6v9TDVrU0CdVGRDA"
  }
}

{  
  "jti": "-BwC3ESc6acc2lTc",
  "htm": "POST",
  "htu": "https://server.example.com/token",
  "iat": "1562262616"
}

Explicitly typed

The public key for which proof-of-possession is being demonstrated

Asymmetric signature algorithms only

Minimal info about the HTTP request (method & URI)

Only valid for a limited time window relative to creation time

Unique identifier for replay checking

Other stuff could go here
Access Token Request

POST /token HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

DPoP: eyJ0eXAiOiJkcG9wK2p3dCI6IkVTMjU2Iiwic3N3ZXVuY3Rpb25zIjoiIiwiaWF0IjoxNTYyMjYyNjE2fQ.

grant_type=authorization_code
&code=SplxlOBeZQQYbYS6WxSbIA
&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
&code_verifier=bEaL42iczC-o-xBk0K2vuJ6U-y1p9r_wW2dFVIWgjz-

DPoP proof JWT in HTTP header
Access Token Response

HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store

{
"access_token":"eyJhbGciOiJFUzI1NiIsImtpZCI6IkJlQUxrYiJ9.eyJzdWIiOiJzb21lbc2l6ZSIsImltZCI6MCwxOTQ2NTQ5MjIzLXNldmVyZXIiLCJ0b2tlblwiOiJ0cmFuc2VhcmQ6UGVyc2lvbi5jbyJ9.eyJqdGkiOiJzdWI6MCIsImlhdCI6MTY0MjcyNTA1NjIwN1wiZXhwIjoxNjM4MTM0NDQ2fQ.0kVigzYhF1MQ",
"token_type":"DPoP",
"expires_in":3600,
"refresh_token":"4LTC8lb0acc6Oy4esc1Nk9BWC0imAwH7kic16BDC2",
}
DPoP Bound Access Token
JWT & Introspection Response

```
{
  "sub": "someone@example.com",
  "iss": "https://server.example.com",
  "aud": "https://resource.example.org",
  "nbf": 1562262611,
  "exp": 1562266216,
  "cnf":
  {
    "jkt": "0ZcOCORZNYy-DWpqq30jZyJGHTN0d2HglBV3uiguA4I"
  }
}
```

Confirmation claim carries the SHA-256 JWK Thumbprint of the DPoP public key to which the access token is bound.
Protected Resource Request

GET /protectedresource HTTP/1.1
Host: resource.example.org
Authorization: DPoP eyJhbGciOiJFUFUzI1NiIsImtpZCI6IkQxYiIiLCJoIjoiU09SVFh0MkZETk9SV1NhQ1FvSUxUS0dUTE5QVJiVUtvQ0RkIiwim6w6NjViYmUyNGQ1YTQwMDIzZjI2MzA3ZTQ5YjM2YjBlZGQ0NTg1NjQ0Y2EzYTBkIiwiY2FkZ2ZcIjoiYjYyZTI2MGNjMjMyYzI1YmZkYjI4ZjBmOTZhN2ViZmI2YzA3ZjFhZjVmZDk1IiwicmF0aWQiOiJkaHlsZS1mb3JtZS1ncmFpZC10aWQiLCJ1c2VyIjoiaHR0cHM6Ly9zZXJ2aWN0LmNsb3Vkcy5jb20ifQ.eyJkZiI6eyJzaXplIjoiY3VycmVuYS1zZXNzaW9uZG93b3Jrc2luZw==fQ.eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJtZXNzYWdlX3R5cGUiOiJodHRwczovL3d3dy5zaXplLmNvbS9oZWlnaHQvIiwiaWRvaWVuX3R5cGUiOiJodHRwczovL3d3dy5zaW1nLmdvb2dvcy9kaXNld3MvIiwibWF4IjoiaHR0cHM6Ly9zZXJ2aWN0LmNsb3Vkcy5jb20ifQ.eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJtZXNzYWdlX3R5cGUiOiJodHRwczovL3d3dy5zaXplLmNvbS9oZWlnaHQvIiwiaWRvaWVuX3R5cGUiOiJodHRwczovL3d3dy5zaW1nLmdvb2dvcy9kaXNld3MvIiwibWF4IjoiaHR0cHM6Ly9zZXJ2aWN0LmNsb3Vkcy5jb20ifQ.

DPoP: eyJ0eXAiOiJkX2IsIm1nIjoiQWRtaW5pbmciLCJza21lIjoiaHR0cHM6Ly9zZXJ2aWN0LmNsb3Vkcy5jb20ifQ.eyJ1c2VyX3N0cyI6ImV4cGlyZWNvbnRzX2VycmVuYWwtY2Vyd2F0aW9uY2F0ZWdvcnkyIiwicGFzc3dvcmkuY29tIiwiY2xvY2F0aW9uX3Jlc3VtZT0iY2F0ZmwifQ.eyJtZXNzYWdlX3R5cGUiOiJodHRwczovL3d3dy5zaXplLmNvbS9oZWlnaHQvIiwiaWRvaWVuX3R5cGUiOiJodHRwczovL3d3dy5zaW1nLmdvb2dvcy9kaXNld3MvIiwibWF4IjoiaHR0cHM6Ly9zZXJ2aWN0LmNsb3Vkcy5jb20ifQ.eyJtZXNzYWdlX3R5cGUiOiJodHRwczovL3d3dy5zaXplLmNvbS9oZWlnaHQvIiwiaWRvaWVuX3R5cGUiOiJodHRwczovL3d3dy5zaW1nLmdvb2dvcy9kaXNld3MvIiwibWF4IjoiaHR0cHM6Ly9zZXJ2aWN0LmNsb3Vkcy5jb20ifQ.eyJkZiI6eyJzaXplIjoiY3VycmVuYS1zZXNzaW9uZG93b3Jrc2luZw==fQ.eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJtZXNzYWdlX3R5cGUiOiJodHRwczovL3d3dy5zaXplLmNvbS9oZWlnaHQvIiwiaWRvaWVuX3R5cGUiOiJodHRwczovL3d3dy5zaW1nLmdvb2dvcy9kaXNld3MvIiwibWF4IjoiaHR0cHM6Ly9zZXJ2aWN0LmNsb3Vkcy5jb20ifQ.eyJkZiI6eyJzaXplIjoiY3VycmVuYS1zZXNzaW9uZG93b3Jrc2luZw==fQ.eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJtZXNzYWdlX3R5cGUiOiJodHRwczovL3d3dy5zaXplLmNvbS9oZWlnaHQvIiwiaWRvaWVuX3R5cGUiOiJodHRwczovL3d3dy5zaW1nLmdvb2dvcy9kaXNld3MvIiwibWF4IjoiaHR0cHM6Ly9zZXJ2aWN0LmNsb3Vkcy5jb20ifQ.eyJkZiI6eyJzaXplIjoiY3VycmVuYS1zZXNzaW9uZG93b3Jrc2luZw==fQ.eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJtZXNzYWdlX3R5cGUiOiJodHRwczovL3d3dy5zaXplLmNvbS9oZWlnaHQvIiwiaWRvaWVuX3R5cGUiOiJodHRwczovL3d3dy5zaW1nLmdvb2dvcy9kaXNld3MvIiwibWF4IjoiaHR0cHM6Ly9zZXJ2aWN0LmNsb3Vkcy5jb20ifQ.

DPoP proof

DPoP public key bound access token
Document History and Status
(and workation slideshow)
They’ll tell the story of tonight

OAuth Security Workshop
Stuttgart*
March 2019

* Took the train from Frankfurt
backstory on the "shiny name"*

*Hannes https://youtu.be/tUmT5qqlKik?t=4178
We’ll always have Prague

- -00 quickly published & presented
- some interest expressed
- just an individual draft (with all the authority thereby bestowed upon it*)

-01/-02 published & presented
interest again expressed
yet remains an individual draft

“… and running code.”
- Node AS - https://github.com/panva/node-oidc-provider
- Go library - https://github.com/pquerna/dpop
- Java JWT library API enhancements - https://bitbucket.org/b_c/jose4j

IETF #105

Vive la Canada!

Montreal
-03 of the individual draft published
  - smaller tokens via “htm”, “htu”, and “jkt” rather than “http_method”, “http_uri”, and “jkt#S256” respectively
  - clarify/fix “jti” uniqueness requirements in DPoP proof
Advance praise for DPoP

“I have a client that is very keen on binding tokens but not so keen on MTLS […] is pushing me quite hard for DPoP”
– anonymous consultant

“lightweight... application level only... existing libraries”
– unnamed speaker at Vancouver Identity Meetup

“interesting work... lot of potential”
– unspecified Identiverse keynote speaker pictured here

“what’s your take on it? To me it seems simple and very sensible... how soon do you think it might actually turn into something real?”
– anonymous colleague

“very simple, very concise”
– unnamed co-author

“very enthusiastic about the new proposal [...] represents a significant advance in OAuth 2.0”
– unnamed mailing list participant
opportunities for further discussion

- Asymmetric cryptography is not super fast
- Threat model and stated objectives are a bit loose
- Specific claims
- ‘jti’ tracking isn’t always as easy as it seems
- Error code(s) and/or metadata
- MTI and/or algorithm discovery/negotiation
Next Steps
Before IETF #107 in Vancouver

Humbly request that the WG consider a call for adoption!