Workgroup: OAUTH
Internet-Draft: draft-richer-oauth-httpsig-00
Published: 21 June 2021
Intended Status: Standards Track
Expires: 23 December 2021
Authors: J. Richer, Ed.
Bespoke Engineering

OAuth Proof of Possession Tokens with HTTP Message Signatures

Abstract

This extension to the OAuth 2.0 authorization framework defines a method for using HTTP Message Signatures to bind access tokens to keys held by OAuth 2.0 clients.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 23 December 2021.

Copyright Notice

Table of Contents

1. Introduction

The OAuth 2.0 framework provides methods for clients to get delegated access tokens from an authorization server for accessing protected resources. The access tokens at the center of OAuth 2.0 can be bound to a variety of different mechanisms, including bearer tokens, mutual TLS, or other presentation mechanisms.

Bearer tokens are simple to implement but also have the significant security downside of allowing anyone who sees the access token to use that token. This extension defines a token type that binds the token to a presentation key known to the client. The client uses HTTP Message Signatures to sign requests using its key, thereby proving its right to present the associated access token.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document contains non-normative examples of partial and complete HTTP messages, JSON structures, URLs, query components, keys, and other elements. Some examples use a single trailing backslash '' to indicate line wrapping for long values, as per [RFC8792]. The \ character and leading spaces on wrapped lines are not part of the value.

2. Token Response

When the client makes an access token request, the AS associates the generated access token with the client's registered key from the client's jwks or jwks_uri field. All presentations of this token at

any RS MUST contain an HTTP message signature as described in Section 3.

A bound access token MUST have a token_type value of httpsig. The response MUST contain a keyid value which indicates the key the client MUST use when presenting the access token Section 3. The value of this keyid field MUST uniquely identify a key from the client's registered key set by its kid value.

```
{
   "access_token": "2340897.34j123-134uh2345n",
   "token_type": "httpsig",
   "keyid": "test-key-rsa-pss"
}
```

[[Editor's note: while this document deals only with using a pre-registered key, it would be possible to have different key binding mechanisms, such as the client presenting an ephemeral key during the token request or the AS generating and assigning a key alongside the token. The WG needs to decide if this is in scope of this document or not. The presentation mechanisms would be the same.]]

3. Presenting an HTTP Message Signature Bound Access Token

The algorithm and key used for the HTTP Message Signature are derived from the client's registered information. The key is taken from the client's registered jwks or jwks_uri field, identified by the keyid field of the token response Section 2. The signature algorithm is determined by the alg field of the identified key, following the method for JSON Web Algorithm selection described in [I-D.ietf-httpbis-message-signatures].

The client MUST include the access token value in an Authorization header using scheme HTTPSig. Note that the scheme value HTTPSig is not case sensitive.

```
Authorization: HTTPSig 2340897.34j123-134uh2345n
```

The client MUST include an HTTP Message Signature that covers, at minimum:

- *The request target of the RS being called
- *The Host header of the RS being called
- *The Authorization header containing the access token value.

The signature parameters MUST include a created signature parameter. The RS SHOULD use this field to ensure freshness of the signed request, appropriate to the API being protected.

The client MUST NOT include an alg signature parameter, since the algorithm is determined by the client's registered key. The client MUST include the keyid signature parameter set to the value returned in the token response Section 2.

In this example, the client has a key with the kid value of test-key-rsa-pss which uses the JWA alg value of PS512. The signature input string is:

```
"@request-target": get /foo
"host": example.org
"authorization": HTTPSig 2340897.34j123-134uh2345n
"@signature-params": ("@request-target" "host" "authorization")\
  ;created=1618884475;keyid="test-key-rsa-pss"
```

This results in the following signed HTTP message, including the access token.

```
GET /foo HTTP/1.1
Host: example.com
Date: Tue, 20 Apr 2021 02:07:55 GMT
Authorization: HTTPSig 2340897.34j123-134uh2345n
Signature-Input: sig1=("@request-target" "host" "authorization")\
  ;created=1618884475;keyid="test-key-rsa-pss"
Signature: sig1=:o+Fy/a6IIWhHwnMFhsHqfXEpheWGBMOU3pheT50zA8rL5F8Nur\
  xBKAPylMGBWYCKH5Bd+TB0Co6vqANlXyOCM9Zr5c/UmR5WGex5/OgJJmfN7gOVOH5\
  pB2Zxa233xsohfwo9liBlctukN5//E3F04rKjIkoeTFJiS+hMcOzn29esgFSEl4Jy\
  oO5Q8snMIsC56ZAPYwU7rJis1Wvl6Y9/9tpW6gIn/SHwArhPQSAb0zZy6mCiw654n\
  CaKw5NYJ9S0DZlnV4T7nJtdZsHOkddF6kH4WVka3ev0xONI5kYkEdR1Gw0VAE9thi\
  p+3/aFoUVTJ/1J6JfehZpXqehwv3KNoQ==:
```

An RS receiving such a signed message and a bound access token MUST verify the HTTP Message Signature as described in [I-D.ietf-httpbis-message-signatures]. The RS MUST verify that all required portions of the HTTP request are covered by the signature by examining the contents of the signature parameters.

[[Editor's note: we should define confirmation methods for access tokens here, including JWT values and introspection response values to allow the RS to verify the signature w/o the client's registration information.]]

4. Acknowledgements

5. IANA Considerations

[[TBD: register the token type and new parameters into their appropriate registries, as well as the JWT and introspection parameters.]]

6. Security Considerations

[[TBD: There are a lot of security considerations to add.]]

All requests have to be over TLS or equivalent as per [BCP195].

7. Privacy Considerations

[[TBD: There are a lot of privacy considerations to add.]]

8. Normative References

[BCP195] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", May 2015, <https://www.rfc-editor.org/info/bcp195>.

[I-D.ietf-httpbis-message-signatures] Backman, A., Richer, J., and M. Sporny, "Signing HTTP Messages", Work in Progress, Internet-Draft, draft-ietf-httpbis-message-signatures-05, 8 June 2021, <https://www.ietf.org/archive/id/draft-ietf-httpbis-message-signatures-05.txt>.

[I-D.ietf-oauth-dpop] Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., and D. Waite, "OAuth 2.0 Demonstrating Proof-of-Possession at the Application Layer (DPoP)", Work in Progress, Internet-Draft, draft-ietf-oauth-dpop-03, 7 April 2021, <https://www.ietf.org/archive/id/draft-ietf-oauth-dpop-03.txt>.

[I-D.ietf-oauth-rar] Lodderstedt, T., Richer, J., and B. Campbell, "OAuth 2.0 Rich Authorization Requests", Work in Progress, Internet-Draft, draft-ietf-oauth-rar-05, 15 May 2021, <https://www.ietf.org/archive/id/draft-ietf-oauth-rar-05.txt>.

[I-D.ietf-oauth-signed-http-request] Richer, J., Bradley, J., and H. Tschofenig, "A Method for Signing HTTP Requests for OAuth", Work in Progress, Internet-Draft, draft-ietf-oauth-signed-http-request-03, 8 August 2016, <https://

www.ietf.org/archive/id/draft-ietf-oauth-signed-http-request-03.txt>.

[I-D.ietf-secevent-subject-identifiers] Backman, A. and M. Scurtescu, "Subject Identifiers for Security Event Tokens", Work in Progress, Internet-Draft, draft-ietf-secevent-subject-identifiers-08, 24 May 2021, <https://www.ietf.org/archive/id/draft-ietf-secevent-subject-identifiers-08.txt>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3230] Mogul, J. and A. Van Hoff, "Instance Digests in HTTP", RFC 3230, DOI 10.17487/RFC3230, January 2002, <https://www.rfc-editor.org/info/rfc3230>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <https://www.rfc-editor.org/info/rfc5646>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/info/rfc6749>.

[RFC6750] Jones, M. and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage", RFC 6750, DOI 10.17487/RFC6750, October 2012, <https://www.rfc-editor.org/info/rfc6750>.

[RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, DOI 10.17487/RFC7234, June 2014, <https://www.rfc-editor.org/info/rfc7234>.

[RFC7468] Josefsson, S. and S. Leonard, "Textual Encodings of PKIX, PKCS, and CMS Structures", RFC 7468, DOI 10.17487/

RFC7468, April 2015, <https://www.rfc-editor.org/info/rfc7468>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <https://www.rfc-editor.org/info/rfc7517>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC8705] Campbell, B., Bradley, J., Sakimura, N., and T. Lodderstedt, "OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens", RFC 8705, DOI 10.17487/RFC8705, February 2020, <https://www.rfc-editor.org/info/rfc8705>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

Appendix A. Document History

*-00

-Initial individual draft.

Author's Address

Justin Richer (editor)
Bespoke Engineering

Email: ietf@justin.richer.org
URI: https://bspk.io/