OAuth Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 20, 2016

N. Sakimura
Nomura Research Institute
K. Li
Alibaba Group

October 18, 2015

Sender Constrained JWT for OAuth 2.0
draft-sakimura-oauth-rjwtprof-06

Abstract

This discussion document describes a method to indicate a sender constraint within JWT. It could potentially be incorporated into Proof-Of-Possession Semantics for JSON Web Tokens(JWTs) [POPS]. This document was created in response to the WGLC of it.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 http://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 April 20, 2016.

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Notational Conventions 3
2. Terms and definitions 3
3. Justification 3
4. Sender Constraint Representation 4
5. Client Authentication 4
6. Finding the client key 5
6.1. URI client ID 5
6.2. pre-shared key tables 5
6.3. Via client metadata API of the authorization server . . . 6
7. IANA Considerations 6
7.1. Named Authentication Scheme 6
8. Security Considerations 6
9. Acknowledgements 6
10. References 7
10.1. Normative References 7
10.2. Informative References 7
Appendix A. Document History 7
Authors' Addresses 7

1. Introduction

OAuth 2.0 Proof-of-Possession (PoP) Security Architecture [POPA] identifies Sender Constraint and Key Confirmation as possible threat mitigation methods against the use of token by an unauthorized presenter. While Proof-Of-Possession Semantics for JSON Web Tokens (JWTs) [POPS] touches briefly on the Sender Constraint, it is only one paragraph within a introductory text and does not discuss it in detail. Instead, it devotes much of the discussion to the Key Confirmation method. It also is making the usage of such token against the resource server out of scope.

This discussion draft describes a way to express the Sender Constraint in the JWT, as well as one possible way of using it to access a protected resource.

The initial draft of this document was created in response to the WGLC of the Proof-Of-Possession Semantics for JSON Web Tokens(JWTs) [POPS].

1.1. Notational Conventions

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 RFC 2119 [RFC2119].

Unless otherwise noted, all the protocol parameter names and values are case sensitive.

2. Terms and definitions

For the purpose of this document, the terms defined in RFC6749 [RFC6749] is used. In addition, following term is defined.

Authorized Presenter Party that the token is intended to be used by.

3. Justification

There are scenarios that the bearer token may be stolen, modified, reused or replayed. To prevent these threats, resource servers need to obtain additional assurance that the client is indeed authorized to present an access token. The detailed use cases can be found in OAuth 2.0 Proof-of-Possession (PoP) Security Architecture [POPA] specification that sites token reuse by the resource server and eavesdropping of the resource request among others. Some additional use-cases such as token leaking from the client's database or authorization server's database is also conceivable.

As described in OAuth 2.0 Proof-of-Possession (PoP) Security Architecture [POPA] specification, there are several ways to prevent these bearer token threats: Confidentiality Protection, Sender Constraint and Key Confirmation. Key Confirmation mechanism is described in OAuth 2.0 Proof-Of-Possession Semantics for JSON Web Tokens (JWTs) [POPS] specification in detail, but Sender Constraint mechanism is not explained in detail.

Sender confirmation mechanism has some advantage in some cases over the general key confirmation mechanism explained in [POPS] in cases such as:

(1) The client's public key is published in a known way in the ecosystem, e.g., in .well-known/jwks and the private key is stored in a HSM.

(2) The resource server wishes to have some non-repudiation of the client.

These can be achieved with relative ease with sender confirmation.

Key Confirmation mechanism is more general in nature. It is applicable even in the case where client's privacy is sought or the client is a public client using OAuth PKCE [PKCE]. As the downside of it, it requires a complete key distribution protocol and can become more complicated. Sender Confirmation mechanism should also be specified, and it can work as an alternative mechanism to mitigate the bearer token threats.

4. Sender Constraint Representation

Sender Constraint is expressed by including the following member at the top level of JWT payload.

azp The Client ID of the Authorized Presenter.

Following is an example of such JWT payload.

```
{
 "iss": "https://server.example.com",
 "sub": "joe@example.com",
 "azp": "https://client.example.org",
 "aud": "https://resource.example.org",
 "exp": "1361398824",
 "nbf": "1360189224",
}
```

Figure 1 Example of Sender Constrained JWT.

5. Client Authentication

The resource server that supports this specification MUST authenticate the Client. In this document a possible method is proposed as follows:

(1). The authorized presenter issues a HEAD or GET request to the resource server.

```
GET /resource/1234 HTTP/1.0
Host: server.example.com
```

(2). The resource server returns a HTTP 401 response with "WWW-Authenticate" header with "Named" scheme, which includes nonce.

```
HTTP/1.0 401 Unauthorized
Server: HTTPd/0.9
Date: Wed, 14 March 2015 09:26:53 GMT
WWW-Authenticate: Named nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093"
```

(3). The client creates JWS compact serialization over the nonce.

(4). The client sends the request to the resource server, this time with Authorization: header with Named scheme and access token and the JWS.

```
GET /resource/1234 HTTP/1.0
Host: server.example.com
Authorization: Named at="access.token.jwt", s="jws.of.nonce"
```

(5). The resource server finds the client key corresponding to the value of "azp" in the access token. It may have been obtained through client registration at the Issuer or through .well-known/jwk etc.

(6). The resource server creates the JWS of the nonce and compares it with the value of "s" of the Authorization header. If it fails, the process stops here and the resource access MUST be denied.

(7). The resource server MUST verify the access token. If it is valid, the resource SHOULD be returned as HTTP response.

6. Finding the client key

When the resource server authenticates the client, it has to find out the keys that corresponds to the signing key of the client. There are several possible ways to do this.

6.1. URI client ID

When the Client ID is a URI, then the key can be found from the .well-known/jwk URI.

6.2. pre-shared key tables

Alternatively, the collection of the keys can be pre-shared among the participants in advance as a key table that lists the client ID - public key pair.

6.3. Via client metadata API of the authorization server

Client Metadata can be exposed through a client metadata API at the Authorization Server, which can be defined by the authorization server in a way similar to OAuth 2.0 Token Introspection [TINTRO].

7. IANA Considerations

7.1. Named Authentication Scheme

A new scheme has been registered in the HTTP Authentication Scheme Registry as follows:

Authentication Scheme Name: Named

Reference: Section 5 of this specification

Notes (optional): The Named Authentication scheme is intended to be used only with OAuth Resource Access, and thus does not support proxy authentication.

8. Security Considerations

To avoid the situation that the client identifier is fake, the resource server that supports this specification MUST authenticate the client.

Integrity protection SHOULD be applied via a keyed message digest or a digital signature, to prevent an adversary from changing any elements conveyed within the JWT payload. Special care MUST be applied when carrying client's secret key inside the JWT, since those not only require integrity protection, but also confidentiality protection. The client's secret key must be encrypted and kept securely.

A client identifier may be used as a correlation handle if it has relationship with the user, e.g. mobile phone number. Thus, for privacy reasons, it is recommended to keep client identifier confidentially protected.

9. Acknowledgements

Thanks Mike Jones for the reviews, and thanks Brian Campbell, John Bradley for the discussions.

10. References

10.1. Normative References

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

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

10.2. Informative References

[PKCE] Sakimura, N., "Proof Key for Code Exchange by OAuth Public Clients", July 2015.

[POPA] Hunt, P., Ed., "OAuth 2.0 Proof-of-Possession (PoP) Security Architecture", March 2015.

[POPS] Jones, M., "Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs)", March 2015.

[TINTRO] Richer, J., "OAuth 2.0 Token Introspection", July 2015.

Appendix A. Document History

-05 Added more justification. Also, added "Finding the client key" section.

-04 Added justification section

-03 Removed most of the duplication with [POPS]

-02 Included key confirmation method etc. The first version on the tools.ietf.org. (Previous versions were sent just as email attachments.)

Authors' Addresses

Nat Sakimura
Nomura Research Institute

Email: sakimura@gmail.com

Kepeng Li
Alibaba Group

Email: kepeng.lkp@alibaba-inc.com