ACE Working Group
Internet-Draft
Intended status: Informational
Expires: May 3, 2018

L. Seitz
RISE SICS
S. Erdtman
Spotify AB

October 30, 2017

Raw-Public-Key and Pre-Shared-Key as OAuth client credentials
draft-erdtman-ace-rpcc-02

Abstract

This document describes Transport Layer Security (TLS) authentication using Raw-Public-Key and Pre-Shared-Key as new mechanisms for OAuth client authentication. Although defined for TLS the mechanisms are equally applicable for DTLS.

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 May 3, 2018.

Copyright Notice

Table of Contents

1. Introduction

This document describes Transport Layer Security (TLS) authentication using Raw-Public-Key and Pre-Shared-Key as the mechanism for OAuth client authentication. Examples of endpoint requiring client authentication are token and introspection.

The OAuth 2.0 Authorization Framework [RFC6749] defines a shared secret method of client authentication but also allows for the definition and use of additional client authentication mechanisms when interacting with the authorization server's token endpoint. This document describes two additional mechanisms of client authentication utilizing Raw-Public-Key [RFC7250] and Pre-Shared-Key TLS [RFC4279], which provide better security characteristics than shared secrets.

To get most bennefits and improved security with these new client credential types it is recomended to use the 'one credential per Client Software Instance' paradigm. This can be achived by letting the client dynamicly register as described in [RFC7591].

1.1. 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].

2. Pre-Shared-Key for Client Authentication

The following section defines, as an extension of OAuth 2.0, Section 2.3 [RFC6749], using Pre-Shared-Key with TLS [RFC4279] to authenticate the client. This method is registered as 'tls_client_psk' in "OAuth Token Endpoint Authentication Methods" registry. If this method is to be used, the client and the Authorization Server MUST share a secret key, and they MUST agree on an identifier for this key.

The (D)TLS handshake MUST be done according to [RFC4279], with the client indicating support for one or more Pre-Shared-Key cipher suites and authorization server selecting a Pre-Shared-Key cipher suite. In order to enable the authorization server to select the correct pre-shared-key the client MUST send the key identifier in the psk-identity field of the ClientKeyExchange message. How the authorization server maps the identifier to a pre-shared-key, and to a specific client is out of scope for this specification.

Note that the key identifier MUST be 2^16 bytes or shorter, in order to fit into the psk-identity field.

3. Raw-Public-Key for Client Authentication

The following section defines, as an extension of OAuth 2.0, Section 2.3 [RFC6749], the use of Raw-Public-Key with (D)TLS [RFC7250] to authenticate the client. This method is registered as 'tls_client_rpk' in "OAuth Token Endpoint Authentication Methods" registry.

The (D)TLS handshake MUST be done according to [RFC7250], with the client indicating support for Raw-Public-Key certificates and the authorization server asking client send its Raw Public Key certificate. Since the client cannot send an explicit client or key identifier in the handshake, the authorization server MUST derive a client identifier from RPK that the client uses.

Note to implementers: Authorization servers can use the following method to map a Raw Public Key to a client identifier: The client identifier is generated from the Raw Public Key using the procedure specified in section 3 of [RFC6920]. The digest is calculated on the Raw Public Key only (not on the SubjectPublicKeyInfo used in the handshake). An example is shown in Figure 1.


```
Raw Public Key (Base64 encoded):
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEEtboxNKPgxEKV9JTNzy
tUvAbxEfkCTVB9kOzheF5wRAoOz2NKP+ln+XLVAQSp1D6jfo09tppvN
poQA1nnBNH6A==";

Encoding:
ni:///sha-256;xzLa24yOBeCkos3VFzD2gd83Urohr9TsXqY9nhdDN0
```

Figure 1: Example encoding of a raw public key in the Named Information URI Format

4. Dynamic Registration

For dynamic registration of a RPK this specification registers the new parameter 'rpk' to the Client Registration Metadata Registry. When used this parameter MUST contain a JSON Web Key representing the public key of the client. When 'rpk' is present in the registration request 'token_endpoint_auth_method' MUST include 'tls_client_rpk'.

For dynamic registration of a PSK this specification registers the new parameter 'psk' to the Client Registration Metadata Registry. When used this parameter MUST contain a JSON Web Key representing the key of the client. When registering the client can include the key in the registrations request or the authorisation can generate the key and return it. If the 'psk' attribute is present in a request 'token_endpoint_auth_method' MUST include 'tls_client_psk'. To request the authorisation server to generate the key the client includes 'tls_client_psk' in 'token_endpoint_auth_method' but does not send 'psk' attribute.

The 'jwks' and 'jwks_uri' is not used to avoid conflict and confusion with application layer keys.

5. Acknowledgements

This document is highly inspired by [I-D.ietf-oauth-mtls] written by B. Campbell, J. Bradley, N. Sakimura and T. Lodderstedt.

6. IANA Considerations

6.1. OAuth Dynamic Client Registration Metadata Registration

This specification requests registration of the following value in the IANA "OAuth Dynamic Client Registration Metadata" registry [IANA.OAuth.Parameters] established by [RFC7591].

6.1.1. Registry Contents

- Client Metadata Name: "rpk"
- Client Metadata Description: JWK for client Raw-Public-Key, can be included in request.
- Change Controller: IESG
- Specification Document(s): [[this specification]]

- Client Metadata Name: "psk"
- Client Metadata Description: JWK for client Pre-Shared-Key, can be included both in request and response.
- Change Controller: IESG
- Specification Document(s): [[this specification]]

6.2. Token Endpoint Authentication Method Registration

This specification requests registration of the following value in the IANA "OAuth Token Endpoint Authentication Methods" registry [IANA.OAuth.Parameters] established by [RFC7591].

6.2.1. Registry Contents

- Token Endpoint Authentication Method Name: "tls_client_rpk"
- Change Controller: IESG
- Specification Document(s): [[this specification]]

- Token Endpoint Authentication Method Name: "tls_client_psk"
- Change Controller: IESG
- Specification Document(s): [[this specification]]

7. Security Considerations

TBD

8. References

8.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4279] Eronen, P., Ed. and H. Tschofenig, Ed., "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, DOI 10.17487/RFC4279, December 2005, <https://www.rfc-editor.org/info/rfc4279>.

[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>.

[RFC6920] Farrell, S., Kutscher, D., Dannewitz, C., Ohlman, B., Keranen, A., and P. Hallam-Baker, "Naming Things with Hashes", RFC 6920, DOI 10.17487/RFC6920, April 2013, <https://www.rfc-editor.org/info/rfc6920>.

[RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7250, DOI 10.17487/RFC7250, June 2014, <https://www.rfc-editor.org/info/rfc7250>.

[RFC7591] Richer, J., Ed., Jones, M., Bradley, J., Machulak, M., and P. Hunt, "OAuth 2.0 Dynamic Client Registration Protocol", RFC 7591, DOI 10.17487/RFC7591, July 2015, <https://www.rfc-editor.org/info/rfc7591>.

8.2. Informative References

[I-D.ietf-oauth-mtls]
Campbell, B., Bradley, J., Sakimura, N., and T. Lodderstedt, "Mutual TLS Profile for OAuth 2.0", draft-ietf-oauth-mtls-04 (work in progress), October 2017.

[IANA.OAuth.Parameters]
IANA, "OAuth Parameters", <http://www.iana.org/assignments/oauth-parameters>.

Authors' Addresses

Ludwig Seitz
RISE SICS
Scheelevaegen 17
Lund 223 70
SWEDEN

Email: ludwig.seitz@ri.se

Samuel Erdtman
Spotify AB
Birger Jarlsgatan 61, 4tr
Stockholm 113 56
Sweden

Email: erdtman@spotify.com