SIP Core
Internet-Draft
Updates: 3261 (if approved)
Intended status: Standards Track
Expires: November 29, 2019

R. Shekh-Yusef, Ed.
Avaya
C. Holmberg
Ericsson
V. Pascual
webrtchacks

May 28, 2019

Third-Party Token-based Authentication and Authorization for Session Initiation Protocol (SIP)
draft-ietf-sipcore-sip-token-authnz-00

Abstract

This document defines a mechanism for SIP, that is based on the OAuth 2.0 and OpenID Connect Core 1.0 specifications, to enable the delegation of the user authentication and SIP registration authorization to a dedicated third-party entity that is separate from the SIP network elements that provide the SIP service.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
   1.1. Terminology 3
   1.2. SIP User Agent Types 3
2. Authentication and Authorization flow 4
   2.1. Overview 4
   2.2. Initial Registration 5
   2.3. Subsequent Requests 6
3. Authorization Header Syntax 6
4. JWT as Authorization Grant 6
5. Security Considerations 6
6. IANA Considerations 6
7. Acknowledgments 6
8. Normative References 7
Authors' Addresses 7

1. Introduction

The SIP protocol [RFC3261] uses the framework used by the HTTP protocol for authenticating users, which is a simple challenge-response authentication mechanism that allows a server to challenge a client request and allows a client to provide authentication information in response to that challenge.

OAuth 2.0 [RFC6749] defines a token based authorization framework to allow clients to access resources on behalf of their user.

The OpenID Connect 1.0 [OPENID] specifications defines a simple identity layer on top of the OAuth 2.0 protocol, which enables clients to verify the identity of the user based on the

authentication performed by a dedicated authorization server, as well as to obtain basic profile information about the user.

This document defines an mechanism for SIP, that is based on the OAuth 2.0 and OpenID Connect Core 1.0 specifications, to enable the delegation of the user authentication and SIP registration authorization to a dedicated third-party entity that is separate from the SIP network elements that provide the SIP service.

1.1. Terminology

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 [RFC2119].

1.2. SIP User Agent Types

[RFC6749] defines two types of clients, confidential and public, that apply to the SIP User Agents.

- Confidential User Agent: is a SIP UA that is capable of maintaining the confidentiality of the user credentials and any tokens obtained using these user credentials.

- Public User Agent: is a SIP UA that is incapable of maintainings the confidentiality of the user credentials and any obtained tokens.

2. Authentication and Authorization flow

This flow is used by a Confidential UA with rich UI to authenticate to an authorization server and to directly obtain tokens to be able to register and get service from the SIP network.

2.1. Overview

The following figure provides a high level view of flow of messages:

```
     UA                            Proxy                            AS
   ---------------------------------------------------------------------
     |                               |                               |
   [00] The UA prompts the user to provides his credentials          |
     |                               |                               |
     | [01] HTTP POST /token         |                               |
     |-------------------------------------------------------------->|
     |                               |                               |
     |         [02] 200 OK {access_token, refresh_token, [id_token]} |
     |<--------------------------------------------------------------|
     |                               |                               |
     |                               |                               |
     | [03] REGISTER with access_token                               |
     |------------------------------>|                               |
     |                               |                               |
     |                               |    [04] HTTP POST /introspect |
     |                               |           {access_token}      |
     |                               |------------------------------>|
     |                               |                               |
     |                               |        [05] 200 OK {metadata} |
     |                               |<------------------------------|
     |                               |                               |
     |                   [06] 200 OK |                               |
     |<------------------------------|                               |
     |                               |                               |
```

In step [00], the UA collects the user's credentials with the AS.

In steps [01] and [02], the UA first contacts the Authorization Server to authenticate the user and obtain tokens to be used to get access to the SIP network.

The tokens returned to the UA depend on the type of server: with an OAuth Authorization Server, the tokens provided are the access token and refresh token. With an OpenID Connect server, an additional ID-Token is returned, which contains the SIP URI of the user. The method used to authenticate the user and obtain these tokens is out of scope for this document.

In step [03], the UA starts the registration process with the SIP proxy by sending a REGISTER request with the access token it obtained previously.

The proxy validates the access token, and if the access token provided by the UA is an opaque token, then then proxy MAY perform an introspection, steps [04] and [05], to obtain more information about the token and its scope, as per [RFC7662]. Otherwise, after the proxy validates the token to make sure it was signed by a trusted entity, it inspects its claims and act upon it.

When the proxy is satisfied with the token, it then replies with the 200 OK to complete the registration process.

2.2. Initial Registration

In step [03], the UA starts the registration process with the SIP proxy by sending a REGISTER request with the access token it obtained previously.

If the access token obtained from the AS is an opaque token, then the UA MUST include an Authorization header field with the Bearer scheme in the request to carry the access token, as epcified in section 3.

If the access token obtained from the AS is a JSON Web Token (JWT) [RFC7519], then the UA MUST include the token and grant type in the body of the request, as specified in section 4.

When the proxy is satisfied with the token, it then replies with the 200 OK to complete the registration process.

2.3. Subsequent Requests

All subsequent requests from the UA MUST include a valid access token. The UA MUST obtain a new access token before the access token expiry period to continue to get service from the system.

3. Authorization Header Syntax

This section describes the syntax of the authorization header with the Bearer scheme.

```
    Authorization = "Authorization" HCOLON "Bearer" LWS
                    "access_token" EQUAL access_token COMMA
                    "token_type" EQUAL token_type *(COMMA auth-param)
    access_token = quoted-string
    token_type = quoted-string
```

4. JWT as Authorization Grant

This section describes the syntax of the body of the request when a JWT is used to authorize the request, as defined in [RFC7523].

```
grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer&assertion=<JWT>
```

5. Security Considerations

TODO

6. IANA Considerations

TODO

7. Acknowledgments

TODO

8. Normative References

[OPENID] Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, "OpenID Connect Core 1.0", February 2014.

[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>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[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>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC7523] Jones, M., Campbell, B., and C. Mortimore, "JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants", RFC 7523, DOI 10.17487/RFC7523, May 2015, <https://www.rfc-editor.org/info/rfc7523>.

[RFC7662] Richer, J., Ed., "OAuth 2.0 Token Introspection", RFC 7662, DOI 10.17487/RFC7662, October 2015, <https://www.rfc-editor.org/info/rfc7662>.

Authors' Addresses

Rifaat Shekh-Yusef (editor)
Avaya
425 Legget Drive
Ottawa, Ontario
Canada

Phone: +1-613-595-9106
EMail: rifaat.ietf@gmail.com

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: christer.holmberg@ericsson.com

Victor Pascual
webrtchacks
Spain

EMail: victor.pascual.avila@gmail.com