Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 11, 2013

J. Richer
The MITRE Corporation

September 07, 2012

A Method of Bearer Token Redelegation and Chaining for OAuth 2
draft-richer-oauth-chain-00

Abstract

This document provides a method for a resource server to present a token that it has received from a client back to its authorization server for the purposes of receiving a derivative token for use on another resource server in order to chain together service requests.

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 March 11, 2013.

Copyright Notice

Table of Contents

1. Service Chaining

The OAuth2 Authorization protocol provides methods for clients to request tokens from authorization servers on behalf of resource owners for use at resource servers. However, there are no provisions for a resource server to act as a client itself for another resource server, a practice known generally as service chaining. Typically, the services involved in the chain are within a single security domain, and with OAuth they would be using a single Authorization Server.

For services using the OAuth2 Bearer token profile, it is possible for anyone holding the token to call any other service that accepts the token. While this is functional, it is bad practice since the token is knowingly being re-used by someone other than the client to which it was issued. Since the same token is used in each step, this approach also does not allow for attenuation of rights as the chain progresses.

Using a new form of grant_type, this specification presents such chained resource servers with an alternative approach that takes advantage of the simplicity and structure of the OAuth protocol by providing a means for any resource server to present the token it has been accessed with back to the authorization server in order to exchange it for a token of equal or lesser strength for use with another resource server. In this way, the original access token which has been delegated to the client can be redelegated to a secondary service.

This approach differs slightly from the Refresh Token described in the OAuth 2 Core. With a Refresh Token, the Client presents the token to the authorization server to get a new Access Token without involving the Resource Owner. With a redelegated Access Token, as described in this document, the Resource Server presents the Access Token which was provided to it by a Client in order to get a secondary Access Token.

1.1. Abbreviations Used In This Document

AS Authorization Server

C Client

RO Resource Owner

RS1 Primary Resource Server, initially called by C on behalf of RO

RS2 Chained Resource Server, called by RS1 to fulfill request from C

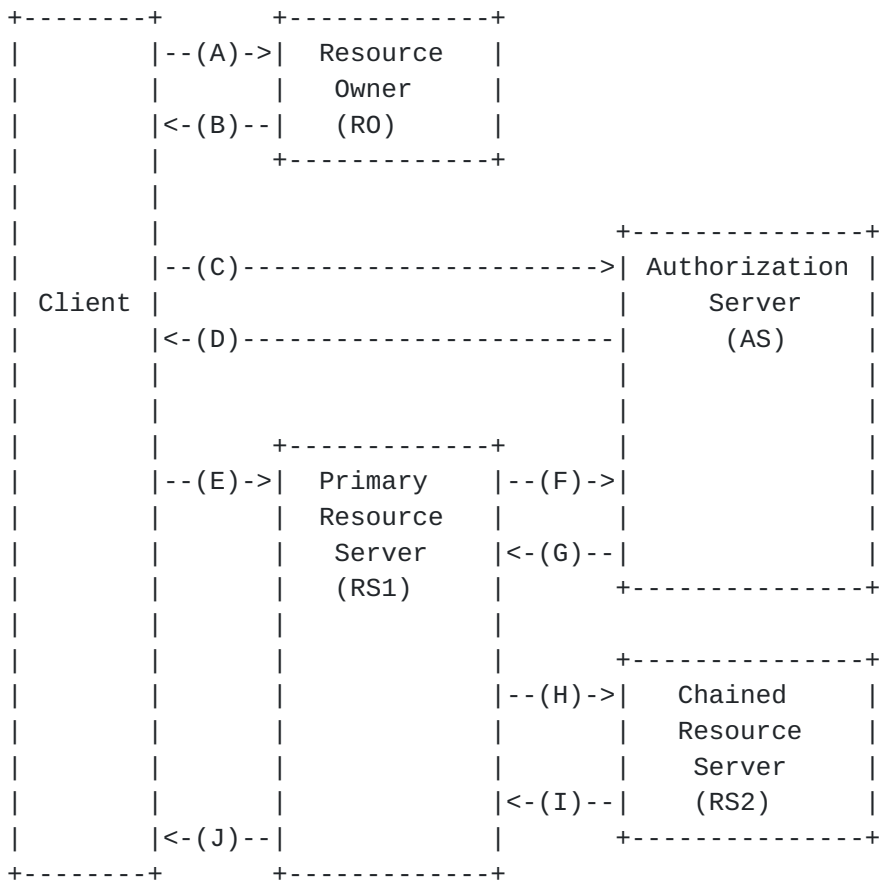
AT1 Bearer Access Token granted by AS to C to access RS1 on behalf of RO

AT2 Bearer Access Token granted by AS to RS1 to access RS2 on behalf of RO

2. Protocol Description

The process begins with any standard OAuth2 protocol flow, where the client obtains AT1 from the AS.

The beginning of the process is standard OAuth2 S.1.2 using any legal OAuth2 grant type to obtain the AT1.



(A) Client requests authorization from Resource Owner

(B) Client receives authorization from the Resource Owner using any valid OAuth2 grant type

(C) Client requests AT1 from the AS by authenticating with the AS and presenting the authorization grant obtained in (B)

(D) AS authenticates the Client and issues access token AT1 for use at RS1

(E) Client presents access token AT1 to RS1 to access a protected resource

(F) RS1 needs to access RS2 to fulfill this request, makes a call to the Token Endpoint on the AS using the redelegate grant_type

(G) AS validates AT1 and issues a token AT2 for use by RS1 against RS2, where the rights assigned to AT2 are a subset of those assigned to AT1

(H) RS1 presents AT2 to RS2 to access a protected resource

(I) RS2 validates token AT2 and returns the protected resource to RS1

(J) Client receives protected resource from RS1, including information sourced from RS2

Steps A-E and J are standard OAuth2 and OAuth2 Bearer tokens involving token AT1. As such, the Client MAY make use of any OAuth2 grant type, such as authorization_code, implicit, client_credentials, password, assertion, or even the redelegation protocol defined in this document. Steps F-G are described in section 3 of this document. Steps H-I are standard OAuth2 Bearer token usage, but using the delegated token AT2.

The means by which the Resource Servers validate the Access Tokens is out of scope of this specification. At the time of this writing, there are two main approaches found in practice: token introspection and structured tokens.

3. Redelegation Grant Type

The Resource Server RS1 makes a request using the Access Token that was presented to it in order to obtain a new Access Token.

3.1. Redelegate Request

To access RS2, RS1 makes a POST request to the Authorization Server's Token Endpoint with the following parameters:

grant_type REQUIRED. Value MUST be set to "urn:ietf:params:oauth:grant_type:redelegate".

token REQUIRED The token that was presented to the resource server by the client, referred to as AT1 in the protocol flow, an OAuth2 Bearer token

scope OPTIONAL a space-separated list of strings as described in OAuth 2. If present, this scope list MUST be equal to or lesser than the scopes incorporated in AT1. The AS MUST issue a token of equal or lesser scope than the token above.

The Authorization Server MAY require RS1 to have registered as a client on its own behalf. In this case, RS1 MUST present its client credentials as described in OAuth2 Core.

3.2. Access Token Response

If the request is valid and authorized, the AS issues an access token, referred to as AT2 in the protocol flow, as described in OAuth2 Core. As this access token is bound to an existing access token, the authorization server MUST NOT issue a refresh token. If the request failed, the authorization server returns an error response as described in OAuth2 Core.

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "access_token":"2YotnFZFEjr1zCsicMWpAA",
  "token_type":"example",
  "expires_in":3600,
  "example_parameter":"example_value"
}
```

3.3. Error Response

If the token request is not valid, such as the access token presented does not allow for redelegation, the AS returns an error response as described in OAuth2 Core.

4. IANA Considerations

[Registration into the OAuth registry for urn:ietf:params:oauth:grant_type:redelegate]

Note to RFC Editor: this section may be removed on publication as an RFC.

5. Security Considerations

A resource server engaging in service chaining and token redelegation SHOULD request a redelegated token with only the minimum set of scopes necessary for calling downstream services.

A resource server MUST indicate in service documentation the full set of scopes required for accessing the full service chain. A redelegation request MUST NOT request escalated privileges without involving the resource owner in a new authorization grant.

6. Acknowledgements

This work has grown from discussions with Paul Nguyen and Stephen Moore, both of MITRE.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

7.2. Informative References

[InfRef] "", 2004.

Appendix A. Standardization of Scopes

The OAuth2 specification explicitly leaves the definition of scopes to the Authorization Server and Protected Resource to agree upon. However, in the course of redelegation, it is sometimes desirable to have a scope value related to the redelegation permission itself. It is RECOMMENDED to use a scope value of "redelegate" if possible.

Author's Address

Justin Richer
The MITRE Corporation
202 Burlington Rd.
Bedford, Massachusetts 01821
USA

Phone: +1-781-271-8176
Fax:
Email: jricher@mitre.org