OAuth Client Credential Flow - Refresh Tokens

Asked 7 years, 8 months ago Modified 3 months ago Viewed 26k times

40

The Scenario

I've recently built an API, and have protected its resources using `OAuth` Bearer Access Tokens.

I've used the `Client_Credentials` Flow, as it will be accessed by clients as opposed to users.

Here's the thing, when a client has successfully provided the `client_id` and the `client_secret` they receive a response like the following :-

```
{
  "access_token": "<Access Token>",
  "token_type": "bearer",
  "expires_in": 1199,
  "refresh_token": "<Refresh Token>"
}
```

Refresh Tokens.

Not knowing much about refresh tokens, i immediately assumed that a client would be able to provide the OAuth Server the `refresh_token` to retrieve a fresh `Access_Token`.

This is 'kind of' correct.

In order to use the `refresh_token` the client still needs to pass the `client_id` and `client_secret` along with the `refresh_token` to get a new access token.

The `grant_type` also needs to be changed to `refresh_token`.

Where is the benefit of a refresh_token using this flow? If I need to pass the client_id and client_secret each time, surely you would just avoid using a refresh token altogether?

oauth-2.0 **oauth**

Share

Improve this question

Follow

edited Feb 27, 2021 at 19:28

Chuck Le Butt

48.7k ● 62 ● 209 ● 297

asked Apr 11, 2017 at 8:28

Derek

8,610 ● 12 ● 58 ● 92

1 This question has nothing specifically to do with C#/ASP.Net and it applicable to anyone building an OAuth2 API. I have edited it to make it more widely applicable. (Great question!) – Chuck Le Butt Feb 27, 2021 at 19:30

3 Answers

Sorted by: Highest score (default)

70

✓

The issuance of a refresh token with the client credential grant has no benefit. That is why the RFC6749 section 4.4.3 indicates `A refresh token SHOULD NOT be included`. Thus its issuance is at the discretion of the authorization server.

From my point of view an authorization server should never issue a refresh token with the client credentials grant:

- With this grant type, the access token is issued *on the first request*. There is no redirection of any kind.
- The client must already store its credentials securely. A refresh token must be stored in the same way. Why bother storing two sensitive pieces of information when only one is needed?

Share Improve this answer Follow

edited Sep 6 at 5:35

answered Apr 11, 2017 at 15:16

Spomky-Labs

16.7k ● 5 ● 44 ● 68

6 Thank you, finally an actual reference to a specification which explains why using refresh token doesn't make sense with client_credentials. We were in doubt, now it's clear!
– Cécile Fecherolle Oct 16, 2020 at 10:53

2 But does it make sense to use refresh tokens in the context of the question: " it will be accessed by clients as opposed to users." clients as in third parties, with no user interaction to renovate refresh/access tokens? – Anonymous May 18, 2021 at 21:09

3 A client can receive access tokens with its credentials. Why would you store and manage refresh tokens when you don’t need them to issue an access token? – Spomky-Labs May 18, 2021 at 21:29

2 @Anonymous I agree. Using refresh tokens are more secure than using ClientId and Secret for retrieving new tokens. – TheLegendaryCopyCoder Jan 25, 2022 at 12:08

2 @TheLegendaryCopyCoder really? If your client credentials are not secure you should really care about it because every single requests to the authorization endpoint require confidential client authentication. In any case, to get a new access token with your refresh token, you will be required to send your client credendials as well. So what's the point storing a refresh token when you can get an access token directly? – Spomky-Labs Jan 25, 2022 at 12:14

1

In Authorization Grant flow Refresh Token act as temporary credentials held by a Client in behalf of the Resource Owner.

Note that a Client can hold multiple refresh tokens for various Resource Owners. This is really important because this is why you need Refresh Tokens. Additionally note that the client must be authenticated as stated in the rfc.

If the client type is confidential or the client was issued client credentials (or assigned other authentication requirements), the client MUST authenticate with the authorization server.

Therefore once trading a Refresh Token for an Access Token the client must authenticate with it's client_id + client_secret (in authorization bearer) + it must send a valid refresh token. Again the Client can do this for multiple resources and for each Resource Owner (and scope) there will be a different Refresh Token (that was previously exchanged by an Authorization Token), nevertheless the Client always authenticates with the same credentials (for the same Authorization Server).

Regarding the Client Credentials flow it specifically states *A refresh token SHOULD NOT be included* as said in top response.

A Refresh Token is not needed because the Client is also the Resource Owner, or at least has full access to the resources granted by the Access Token. That way it is enough just to authenticate with client id and client secret. You could state that that you cannot revoke the Refresh Token and that might be the only case a Refresh Token would be useful. Nevertheless some Authorization Server still return a Refresh Token on the Client Credentials Grant.

Share Improve this answer Follow

answered Nov 7, 2023 at 23:16

fly

57 ● 1 ● 8

▲

-5

▼

The benefit is that he request token normally has a much longer life span than the access token.

Access token is used in communicating with the resource server. Request token is used when communicating with the authorization server.

You could read this as that you may be authorized but that the exact extend of your authorization needs to be reevaluated from time to time. So request token has it use.

Share Improve this answer Follow

answered Apr 11, 2017 at 10:55

Khedron Wilk
5 ● 2

4 By "request token", do you mean *refresh* token?
– gsmendoza May 23, 2019 at 6:44