The OAuth 2.0 Protocol: Bearer Tokens

Based on **http://tools.ietf.org/html/draft-ietf-oauth-v2-bearer-15** (v.15) (December 18, 2011)

Abstract

This specification describes how to use bearer tokens in HTTP requests to access OAuth 2.0 protected resources. Any party in possession of a bearer token (a “bearer”) can use it to get access to the associated resources (without demonstrating possession of a cryptographic key). To prevent misuse, bearer tokens need to be protected from disclosure in storage and in transport.

(v.15)

Table of Contents

2. Authenticated Requests

This section defines three methods of sending bearer access tokens in resource requests to resource servers. Clients MUST NOT use more than one method to transmit the token in each request.

(v.15)

2.1. Authorization Request Header Field

When sending the access token in the “Authorization” request header field defined by HTTP/1.1, Part 7 [I-D.ietf-httpbis-p7-auth], the client uses the “Bearer” authentication scheme to transmit the access token.

For example:

```
GET /resource HTTP/1.1
Host: server.example.com
Authorization: Bearer vF9dft4qmT
```

The “Authorization” header field uses the framework defined by HTTP/1.1, Part 7 [I-D.ietf-httpbis-p7-auth] as follows:

```
credentials = "Bearer" 1*SP b64token
```

The b64token syntax was chosen over the alternative #auth-param syntax also defined by HTTP/1.1, Part 7 [I-D.ietf-httpbis-p7-auth] both for simplicity and for compatibility with existing implementations. If additional parameters are needed in the future, a different scheme would need to be defined.

Clients SHOULD make authenticated requests with a bearer token using the “Authorization” request header field with the “Bearer” HTTP authorization scheme. Resource servers MUST support this method.

(v.15)

3. The WWW-Authenticate Response Header Field

If the protected resource request does not include authentication credentials or does not contain an access token that enables access to the protected resource, the resource server MUST include the HTTP “WWW-Authenticate” response header field; it MAY include it in response to other conditions as well. The “WWW-Authenticate” header field uses the framework defined by HTTP/1.1, Part 7 [I-D.ietf-httpbis-p7-auth] as follows:

```
challenge       = "Bearer" [ 1*SP 1#param ]

param           = realm / scope /
                  error / error-desc / error-uri /
                  auth-param
```

```
scope           = "scope" "=" quoted-string
error           = "error" "=" quoted-string
error-desc      = "error_description" "=" quoted-string
error-uri       = "error_uri" "=" quoted-string
```

A “realm” attribute MAY be included to indicate the scope of protection in the manner described in HTTP/1.1, Part 7 [I-D.ietf-httpbis-p7-auth]. The “realm” attribute MUST NOT appear more than once. The “realm” value is intended for programmatic use and is not meant to be displayed to end users.

The “scope” attribute is a space-delimited list of scope values indicating the required scope of the access token for accessing the requested resource. In some cases, the “scope” value will be used when requesting a new access token with sufficient scope of access to utilize the protected resource. The “scope” attribute MUST NOT appear more than once. The “scope” value is intended for programmatic use and is not meant to be displayed to end users.

If the protected resource request included an access token and failed authentication, the resource server SHOULD include the “error” attribute to provide the client with the reason why the access request was declined. The parameter value is described in ***Section 3.1***. In addition, the resource server MAY include the “error_description” attribute to provide developers a human-readable explanation that is not meant to be displayed to end users. It also MAY include the “error_uri” attribute with an absolute URI identifying a human-readable web page explaining the error. The “error”, “error_description”, and “error_uri” attributes MUST NOT appear more than once.

Producers of “scope” strings MUST NOT use characters outside the set %x21 / %x23-5B / %x5D-7E for representing the scope values and %x20 for the delimiter. Producers of “error” and “error_description” strings MUST NOT use characters outside the set %x20-21 / %x23-5B / %x5D-7E for representing these values. Producers of “error-uri” strings MUST NOT use characters outside the set %x21 / %x23-5B / %x5D-7E for representing these values. Furthermore, “error-uri” strings MUST conform to the URI-Reference syntax. In all these cases, no character quoting will occur, as senders are prohibited from using the %5C (‘) character.

For example, in response to a protected resource request without authentication:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer realm="example"
```

And in response to a protected resource request with an authentication attempt using an expired access token:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer realm="example",
                  error="invalid_token",
                  error_description="The access token expired"
```

(v.15)

3.1. Error Codes

When a request fails, the resource server responds using the appropriate HTTP status code (typically, 400, 401, 403, or 405), and includes one of the following error codes in the response:

invalid_request

The request is missing a required parameter, includes an unsupported parameter or parameter value, repeats the same parameter, uses more than one method for including an access token, or is otherwise malformed. The resource server SHOULD respond with the HTTP 400 (Bad Request) status code.

invalid_token

The access token provided is expired, revoked, malformed, or invalid for other reasons. The resource SHOULD respond with the HTTP 401 (Unauthorized) status code. The client MAY request a new access token and retry the protected resource request.

insufficient_scope

The request requires higher privileges than provided by the access token. The resource server SHOULD respond with the HTTP 403 (Forbidden) status code and MAY include the “scope” attribute with the scope necessary to access the protected resource.

If the request lacks any authentication information (i.e. the client was unaware,authentication is necessary or attempted using an unsupported authentication method), the resource server SHOULD NOT include an error code or other error information.

For example:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer realm="example"
```

(v.15)