OAuth Access Tokens using credentials
draft-dehora-farrell-oauth-accessstoken-creds-01.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 10, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

OAuth Access Tokens using credentials is a technique for allowing user agents to obtain an OAuth access token on behalf of a user without requiring user intervention or HTTP redirection to a browser. OAuth itself is documented in the OAuth Core 1.0 Specification.
Editorial Note

To provide feedback on this Internet-Draft, email the authors.

Table of Contents

1. Introduction ........................................... 4
2. Terminology ........................................... 5
3. Applicability .......................................... 6
4. Client request to obtain an Access token ............... 7
   4.1. Request ........................................... 7
   4.2. Response ......................................... 7
   4.3. Accessing Protected Resources .................... 8
5. Security Considerations ................................ 9
6. IANA Considerations .................................... 10
7. Normative References ................................... 11
Appendix A. Revision History ................................ 12
Authors’ Addresses ....................................... 13
1. Introduction

The [OAUTH] Specification is a protocol that enables websites or applications to access protected web resources via an API, without requiring users to disclose their credentials. This draft defines a technique for allowing a user to provide their credentials in cases where HTTP redirection to a browser is unavailable or unsuitable, such as intermediary aggregators and mobile or settop devices.
2. 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].

- **Access Token** - As defined by [OAUTH], a value used by the Consumer to gain access to the Protected Resources on behalf of the User, instead of using the User’s Service Provider credentials.

- **Service Provider** - As defined by [OAUTH], a web application that allows access via OAuth.

- **User** - As defined by [OAUTH], an individual who has an account with the Service Provider.

- **Consumer** - As defined by [OAUTH], a website or application that uses OAuth to access the Service Provider on behalf of the User.

- **Protected Resource(s)** - As defined by [OAUTH], data controlled by the Service Provider, which the Consumer can access through authentication.

- **Consumer Key** - As defined by [OAUTH], a value used by the Consumer to identify itself to the Service Provider.

- **Consumer Secret** - As defined by [OAUTH], a secret used by the Consumer to establish ownership of the Consumer Key.

- **Request Token** - As defined by [OAUTH], a value used by the Consumer to obtain authorization from the User, and exchanged for an Access Token.

- **Token Secret** - A secret used by the Consumer to establish ownership of a given Token.

- **OAuth Protocol Parameters** - Parameters with names beginning with oauth.
3. Applicability

This scheme is intended for use where one or both of the following situations apply:

- the User is using a device that cannot play the HTTP re-direct game normally played in the "3-legged" OAuth model

- the Consumer is an aggregator that will in any case, be presented with the credentials of the end-user

If neither of the above apply, then this specification SHOULD NOT be used.

In addition, the security considerations below MUST be followed, in particular the requirement that communications between the Consumer and Service Provider that contain the user’s credentials MUST be sent via a confidential and mutually authenticated channel. That channel can be provided either via mutually-authenticated transport layer security or a virtual private network providing equivalent security functionality. See the security considerations section below for details.

Once the Access Token has been acquired by the Consumer, then the security requirements of standard OAuth apply.
4. Client request to obtain an Access token

4.1. Request

To request an Access Token in this model, the Consumer makes an HTTP request to the Service Provider’s Access Token URL. The authentication request contains the following parameters:

- x_auth_username - the login credential of the User the client is obtaining a token on behalf of.
- x_auth_password - the pass credential of the User the client is obtaining a token on behalf of.
- x_auth_mode - this value must "client_auth" (referring to the process described here)
- oauth_consumer_key - as defined by [OAUTH].
- oauth_signature_method - as defined by [OAUTH].
- oauth_signature - as defined by [OAUTH]
- oauth_timestamp - as defined by [OAUTH]
- oauth_nonce - as defined by [OAUTH]
- oauth_version - the client MAY send this parameter. If present, value MUST be 1.0. Service Providers MUST assume the protocol version to be 1.0 if this parameter is not present.

The above parameters are contained in the HTTP Authorisation header or as URL parameters. Parameter names and values must be "percent-encoded" to handle characters in different character sets. The request SHOULD use HTTP POST.

4.2. Response

To grant an access token, the Service Provider MUST ensure that:

- The request signature has been successfully verified as per [OAUTH].
- A request with the supplied timestamp and nonce has never been received before.
- The supplied username and password match a User’s credentials.
If successful, the Service Provider generates an Access Token and Token Secret using a 200 Ok response and returns them in the HTTP response body. The response contains the following parameters:

- oauth_token - The Access Token.
- oauth_token_secret - The Token Secret.
- x_auth_expires - a timestamp, in seconds since 1970-01-01T00:00, at which the Access Token expires, or 0 if no expiry is specified.
- Additional parameters - Any additional parameters, as defined by the Service Provider.

### 4.3. Accessing Protected Resources

After successfully receiving the Access Token and Token Secret, the Consumer is able to access the Protected Resources on behalf of the User as per section 7 of [OAUTH]. In other words the Access Token obtained here is no different in capability to the Access Token specified by [OAUTH]. Once authenticated using the above process, the Consumer will sign all subsequent requests for the User’s Protected Resources using the returned Token Secret.
5. Security Considerations

The authentication technique described here is based on HTTP and thus subject to the security considerations found in Section 15 of [RFC2616].

Sending a user name and password pair is contrary to the idea in [OAUTH] that a Consumer will not know the User’s credentials. However without some way to transmit the credentials, there is no way to utilise [OAUTH] in scenarios where redirects to the Service Provider cannot be performed dynamically.

When acquiring an Access Token via this scheme, the relevant communications between the Consumer and Service Provider MUST be strongly protected via a mutually authenticated and confidential channel. Such a channel can be provided via the use of mutually authenticated Transport Layer Security (TLS) [RFC5246] or an equivalent lower layer virtual private network (VPN), for example a tunnel-mode VPN based on IPsec. [RFC4301]

When HTTP is used over TLS, the conventions in [RFC2818] MUST be followed.

Service Providers are advised to respond to unauthorized or unauthenticated requests using an appropriate 4xx HTTP response code (e.g., 401 "Unauthorized" or 403 "Forbidden") in accordance with [RFC2617].
6. IANA Considerations

No IANA actions are required by this document.
7. Normative References


Appendix A. Revision History

version-00: initial draft.

version-01: added applicability statement and increased level of security required
Authors’ Addresses

Bill de hOra

Email: bill@dehora.net
URI:  http://dehora.net/

Stephen Farrell
NewBay Software

Email: sfarrell@newbay.com
URI:  http://www.newbay.com