CHAPTER 2

Authorization and Authentication

Cisco WebEx Social API requests must come through an authorized API consumer and be issued by an authenticated Cisco WebEx Social user. The Cisco WebEx Social API uses the Open Authorization and the xAuth protocols for such authorization and authentication.

This chapter describes the use of OAuth and xAuth with the Cisco WebEx Social API. It includes these topics:

Overview

OAuth is an open authentication protocol that enables an API consumer (an application that enables the API to access Cisco WebEx Social resources and entities) to act on your behalf without you needing to share your login credentials. With OAuth, requests to access protected Cisco WebEx Social resources are signed to verify the authenticity of the request and that you have permission to access the resources. For example, if you execute an API operation to access the resource that represents the community with the identifier 123, the operation returns that resource only if you are an authenticated user with permission to access that community, and OAuth enables this authentication.

Cisco WebEx Social uses the 3-legged OAuth process, which involves the Cisco WebEx Social server, an API consumer, and a user.

The xAuth protocol is a modification of OAuth and provides an alternate method for trusted devices and applications to access Cisco WebEx Social data.

For additional information about OAuth and the request parameters that this chapter describes, see the OAuth Core 1.0 Revision A specification, which is available online.

To determine which authentication protocol is supported for your deployment, consult your system administrator.

OAuth

OAuth provides these general functions:

- API consumer authorization, which allows an API consumer to access Cisco WebEx Social
- User authentication, which authenticates an API user to Cisco WebEx Social

OAuth uses two sets of credentials. One set identifies an API consumer that executes the operations. The other set identifies the Cisco WebEx Social resource owners.

The OAuth authorization flow consists of these steps, which this chapter describes in detail:

1. Add API Consumer—Register an API consumer with Cisco WebEx Social and receive a Consumer Key and a Consumer Secret.
2. Get Request Token—Obtain a temporary token, called a *Request Token*, to be used in the Get User Authorization authorization step.
3. Get User Authorization—Provide the Request Token as a query parameter and receive a verification code for the Request Token.
4. Exchange Request Token for Access Token—Obtain an Access Token in exchange for the Request Token. The Access Token provides authorization for the API consumer to access Cisco WebEx Social data.
5. Refresh Access Token (optional)—Obtain a new Access Token if the current one expires.

xAuth

You can use the Cisco WebEx Social API with a trusted device, such as a mobile client, or with a trusted application. A device or application is considered to be *trusted* when it maintains your Cisco WebEx Social login credentials. To use a trusted device or application to access Cisco WebEx Social data via the API, you can use a modified version of OAuth, called xAuth, to provide authentication.The xAuth flow bypasses the Get Request Token and the Get User Authorization steps, which OAuth uses, and directly exchanges Cisco WebEx Social user credentials for an Access Token and Token Secret.

Detailed OAuth Authorization Flow for the Cisco WebEx Social API

The following sections provide detailed information about the flow of the 3-legged OAuth 1a authorization process that is implemented for the Cisco WebEx Social API:

- Authorization Flow Step 1: Add an API Consumer, page 2-3
- Authorization Flow Step 2: Get Request Token, page 2-3
- Authorization Flow Step 3: Get User Authorization, page 2-5
- Authorization Flow Step 4: Exchange Request Token for Access Token, page 2-6
- Authorization Flow Step 5: Refresh Access Token, page 2-8

Note For additional information about requirements for specific request parameters that the following sections describe, see the OAuth Core 1.0 Revision A specification, which is available online.

Authorization Flow Step 1: Add an API Consumer

Before a Cisco WebEx Social API consumer can successfully execute an API request, the API consumer must sign up with Cisco WebEx Social and provide information about itself. This process *registers* the API consumer with Cisco WebEx Social.

To perform this process, you execute the Add an API Consumer resource operation, in which you provide a set of information with the request. For information about executing this operation, see the "Add an API Consumer" section on page 11-8.

After the API consumer is registered, you receive a Consumer Key and a Consumer Secret. The Consumer Key identifies the API consumer to Cisco WebEx Social. The Consumer Key and Consumer Secret are used when you request a Request Token.

Authorization Flow Step 2: Get Request Token

After an API consumer is registered with Cisco WebEx Social, it must provide the Consumer Key and Consumer Secret that it received to obtain a Request Token. The Request Token is a temporary token that is used in the Get User Authorization.

To obtain a Request Token, you execute the Get Request Token operation, in which you provide the Consumer Key, Consumer Secret, and additional information.

Request

HTTP Method	URI
GET POST	**http://***server*[**:***port_number*]**/quadopen/oauth/request_token** where: • *server*—Host name or IP address of the Cisco WebEx Social server. • *port_number*—Cisco WebEx Social server port number that is used for communication with the API consumer. Must be provided if the port number is not 80.

Request Parameters

A Get Request Token request includes the following request parameters:

Request Parameter	Description
oauth_consumer_key	Consumer Key that was received as described in the "Authorization Flow Step 1: Add an API Consumer" section on page 2-3.
oauth_nonce	Unique random string that the API consumer provides to allow the Cisco WebEx Social server to verify that the request has not yet been made.
oauth_signature	Signature that verifies the request. Use the Consumer Secret to generate this signature. (The Consumer Secret was received as described in the "Authorization Flow Step 1: Add an API Consumer" section on page 2-3.)
oauth_signature_method	Signing algorithm that is used generate the signature. Must be either **PLAINTEXT** or **HMAC-SHA1**.

Request Parameter	Description
oauth_timsetamp	Timestamp that indicates when this request is submitted. This value must be specified in Unix time, which is the number of seconds since January 1, 1970 00:00:00 GMT, excluding leap seconds.
oauth_version	OAuth version being used. Must be **1.0**.
oauth_callback	Callback URL to which Cisco WebEx Social redirects after the user receives authorized access to Cisco WebEx Social data.

The request parameters can be placed in any of these ways:

- In the authorization header of a GET or POST request. In this case, use **Authorization: OAuth** in the header.
- As the body of a POST request. In this case, use **Content-Type: application/x-www-form-urlencoded** in the header.
- As query parameters in a request. The first query parameter must be preceded by a question mark (?). Separate each query parameter with an ampersand (&).

The following example shows a Get Request Token request that uses query parameters in a GET request:

```
GET http://api.quad.cisco.com/quadopen/oauth/request_token
?oauth_consumer_key=aoi1794g9713987
&oauth_nonce=j2093874jajsl39glj39pjwh039098q
&oauth_signature=aj07%saldkj3nlkn%flkenagie16
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1181287363
&oauth_version=1.0
&oauth_callback="http://yoursite.com/callback"
```

Response

The response to a Get Request Token request includes the following response parameters:

Response Parameter	Description
oauth_token	Request Token.
oauth_token_secret	Token Secret, which is associated with the Request Token.
oauth_callback_confirmed	Confirms that you are using Version 1.0a of the OAuth protocol. Always returns **true**.
xoauth_user_auth_url	URL of the Cisco WebEx Social Get User Authorization page.

The following example shows a response to a Get Request Token request:

```
oauth_token=aheoiasyslk&
oauth_token_secret=hjsiwaojshyh&
oauth_callback_confirmed=true&
xoauth_user_auth_url=http%3A%2Fapi.quad.cisco.com%2Foauth%2Fuser_auth%3Foauth_token
    %3Daheoiasyslk
```

Authorization Flow Step 3: Get User Authorization

After an API consumer receives a Request Token, the API requires you to authenticate yourself with Cisco WebEx Social before continuing. After you are authenticated, you are prompted to authorize the API consumer to access Cisco WebEx Social data.

To obtain user authorization, you execute the Get User Authorization operation, in which you provide the Request Token as a query parameter and receive a verification code (oauth_verifier) for the Request Token.

Request

HTTP Method	URI
GET POST	**http://***server*[**:***port_number*]**/quadopen/oauth/user_auth** where: • *server*—Host name or IP address of the Cisco WebEx Social server. • *port_number*—Cisco WebEx Social server port number that is used for communication with the API consumer. Must be provided if the port number is not 80.

Request Parameters

A Get User Authorization request includes the following request parameter:

Request Parameter	Description
oauth_token	Request Token that was received as described in the "Authorization Flow Step 2: Get Request Token" section on page 2-3.

This request parameter must be included in the HTTP as a query parameter in the request. Precede this parameter with a question mark (?).

The following example shows a Get User Authorization request:

```
GET http://api.quad.cisco.com/oauth/user_auth?oauth_token=pwiajshs
```

Response

When you log in to Cisco WebEx Social, you are prompted to authorize the API consumer that you are using. After you do so, you receive a response that includes the following response parameters, which are appended to the callback URL:

Response Parameter	Description
oauth_token	Request Token that was received as described in the "Authorization Flow Step 2: Get Request Token" section on page 2-3.
oauth_verifier	Verification code that is associated with the Request Token. This code and the Request Token will be exchanged for an Access Token. The verification code expires at the same time as its associated Request Token.

The following example shows a response to a Get User Authorization request:

```
Location: http://www.cisco.com?oauth_token=ajeoie86elns&oauth_verifier=9873072\r\n
```

A particular Request Token cannot be used for more than one login ID and Access Token request.

Authorization Flow Step 4: Exchange Request Token for Access Token

After an API consumer is authorized to access Cisco WebEx Social data, the API consumer exchanges the Request Token for an Access Token. The Access Token provides authorization for the API consumer to access Cisco WebEx Social data.

To obtain an Access Token, you execute the Exchange Request Token for Access Token operation, in which you provide the Request Token and related information.

Request

HTTP Method	URI
GET POST	**http://***server*[**:***port_number*]**/quadopen/oauth/access_token** where: • *server*—Host name or IP address of the Cisco WebEx Social server. • *port_number*—Cisco WebEx Social server port number that is used for communication with the API consumer. Must be provided if the port number is not 80.

Request Parameters

An Exchange Request Token for Access Token request includes the following request parameters:

Request Parameter	Description
oauth_consumer_key	Consumer Key that was received as described in the "Authorization Flow Step 1: Add an API Consumer" section on page 2-3.
oauth_nonce	Unique random string that the API consumer provides to allow the Cisco WebEx Social server to verify that the request has not yet been made.
oauth_signature	Signature that verifies the request. Use the Consumer Secret and Token Secret concatenated with an ampersand (&) to generate this signature. (The Consumer Secret was provided as described in the "Authorization Flow Step 1: Add an API Consumer" section on page 2-3. The Token Secret was provided as described in the "Authorization Flow Step 2: Get Request Token" section on page 2-3). If you are using the PLAINTEXT signature method, use an encoded & (%26) to concatenate this string. If you are using HMAC-SHA1, use an unencoded &.
oauth_signature_method	Signing algorithm that is used generate the signature. Must be either **PLAINTEXT** or **HMAC-SHA1**.

Request Parameter	Description
oauth_timestamp	Timestamp that indicates when this request is submitted. This value must be specified in Unix time, which is the number of seconds since January 1, 1970 00:00:00 GMT, excluding leap seconds.
oauth_token	Request Token that was received as described in the "Authorization Flow Step 2: Get Request Token" section on page 2-3.
oauth_version	OAuth version being used. Must be **1.0**.
oauth_verifier	Verification code that is associated with the Request Token. This code was received as described in the "Authorization Flow Step 3: Get User Authorization" section on page 2-5.

The request parameters can be placed in any of these ways:

- In the authorization header of a GET or POST request. In this case, use **Authorization: OAuth** in the header.
- As the body of a POST request. In this case, use **Content-Type: application/x-www-form-urlencoded** in the header.
- As query parameters in a request. The first query parameter must be preceded by a question mark (?). Separate each query parameter with an ampersand (&).

The following example shows an Exchange Request Token for Access Token request that uses query parameters in a GET request:

```
GET http://api.quad.cisco.com/quadopen/oauth/access_token
?oauth_consumer_key=aoi1794g9713987
&oauth_nonce=jw398adkh389wesd8w3knsg
&oauth_signature=aj07%saldkj3nlkn%flkenagie16
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1202709273
&outh_token=pwiajshs
&oauth_version=1.0
&oauth_verifier=9873qnsa38s
```

Response

The response to an Exchange Request Token for Access Token request includes the following response parameters:

Response Parameter	Description
oauth_token	Access Token, which provides the API consumer with access to Cisco WebEx Social data.
oauth_token_secret	Secret associated with the Access Token.
oauth_session_handle	Credential used to refresh an expired valid Access Token.

Response Parameter	Description
quad_openapi_baseurl	URL-encoded base URL of the Cisco WebEx Social API operations, in the following format: **http://***server*[**:***port_number*]**/api/quad/rest** where: • *server*—Host name or IP address of the Cisco WebEx Social server. • *port_number*—Cisco WebEx Social server port number that is used for communication with the API consumer. Must be provided if the port number is not 80.
xoauth_user_id	Cisco WebEx Social User identifier that is associated with this Access Token.

The following example shows a response to an Exchange Request Token for Access Token request:

```
oauth_token=kjhw83akjKJBE098mlklm09uNW&
oauth_token_secret=jaonqzc&
oauth_session_handle=dshj987qNLKNa3987nskSszn987LKJvlk&
quad_openapi_baseurl=http%3A%2F%2Fwebexsocial%2Fapi%2Fquad%2Frest&
xoauth_user_id=190206987
```

Authorization Flow Step 5: Refresh Access Token

By default, an Access Token expires 60 minutes after it is issued. If an Access Token expires, you can obtain a new one so that you can continue your API session. This process is called *refreshing* an Access Token.

To refresh an Access Token, you execute the Refresh Access Token operation, in which you provide the expired Access Token and related information.

Request

HTTP Method	URI
GET POST	**http://***server*[**:***port_number*]**quadopen/oauth/refresh_access_token** where: • *server*—Host name or IP address of the Cisco WebEx Social server. • *port_number*—Cisco WebEx Social server port number that is used for communication with the API consumer. Must be provided if the port number is not 80.

Request Parameters

A Refresh Access Token request includes the following request parameters:

Request Parameter	Description
oauth_consumer_key	Consumer Key that was received as described in the "Authorization Flow Step 1: Add an API Consumer" section on page 2-3. (This Consumer Key also was provided as described in the "Authorization Flow Step 4: Exchange Request Token for Access Token" section on page 2-6.)
oauth_nonce	Unique random string that the API consumer provides to allow the Cisco WebEx Social server to verify that the request has not yet been made.
oauth_signature	Signature that verifies the request. Use the Consumer Secret and Token Secret concatenated with an ampersand (&) to generate this signature. (The Consumer Secret was provided as described in the "Authorization Flow Step 1: Add an API Consumer" section on page 2-3. The Token Secret was provided as described in the "Authorization Flow Step 2: Get Request Token" section on page 2-3). If you are using the PLAINTEXT signature method, use an encoded & (%26) to concatenate this string. If you are using HMAC-SHA1, use an unencoded &.
oauth_signature_method	Signing algorithm that is used generate the signature. Must be either **PLAINTEXT** or **HMAC-SHA1**.
oauth_timestamp	Timestamp that indicates when this request is submitted. This value must be specified in Unix time, which is the number of seconds since January 1, 1970 00:00:00 GMT, excluding leap seconds.
oauth_token	Request Token that was received as described in the "Authorization Flow Step 2: Get Request Token" section on page 2-3.
oauth_session_handle	Credential used to refresh an expired valid Access Token. This credential was received as described in the "Authorization Flow Step 4: Exchange Request Token for Access Token" section on page 2-6.

The request parameters can be placed in any of these ways:

- In the authorization header of a GET or POST request. In this case, use **Authorization: OAuth** in the header.
- As the body of a POST request. In this case, use **Content-Type: application/x-www-form-urlencoded** in the header.
- As query parameters in a request. The first query parameter must be preceded by a question mark (?). Separate each query parameter with an ampersand (&).

The following example shows a Refresh Access Token request that uses query parameters in a GET request:

```
GET http://api.quad.cisco.com/quadopen/oauth/refresh_access_token
?oauth_consumer_key=aoi1794g9713987
&oauth_nonce=mbeoih987234nlkeoi099
&oauth_session_handle=dshj987qNLKNa3987nskSszn987LKJvlk&
&oauth_signature=aj07%saldkj3nlkn%flkenagie16
```

```
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1201259376
&outh_token=pwiajshs
```

Response

The response to a Refresh Access Token request includes the following response parameters:

Response Parameter	Description
oauth_token	Access Token, which provides the API consumer with access to Cisco WebEx Social data
oauth_token_secret	Secret associated with the Access Token
oauth_session_handle	Credential used to refresh an expired valid Access Token
quad_openapi_baseurl	URL-encoded base URL of the Cisco WebEx Social API operations, in the following format: **http://***server*[**:***port_number*]**/api/quad/rest** where: • *server*—Host name or IP address of the Cisco WebEx Social server. • *port_number*—Cisco WebEx Social server port number that is used for communication with the API consumer. Must be provided if the port number is not 80.
xoauth_user_id	Cisco WebEx Social User identifier that is associated with this Access Token

The following example shows a response to a Refresh Access Token request:

```
oauth_token=kjhw83akjKJBE098mlklm09uNW&
oauth_token_secret=jaonqzc&
oauth_session_handle=dshj987qNLKNa3987nskSszn987LKJvlk&
quad_openapi_baseurl=http%3A%2F%2webexsocial%2Fapi%2Fquad%2Frest&
xoauth_user_id=190206987
```

Using an OAuth Access Token in API Requests

After an API consumer receives an Access Token, the API consumer can use the Access Token to access Cisco WebEx Social data.

Cisco WebEx Social supports PLAINTEXT or HMAC-SHA1 signing types for OAuth API access. When passing PLAINTEXT values, Cisco recommends that you use SSL.

The signature base string is the Consumer Secret and Token Secret concatenated with an ampersand (&). For example, assume that you want to execute an API operation to obtain information about posts. Also assume that the API consumer has registered with Cisco WebEx Social and has obtained the Consumer Key abcd1234 and the Consumer Secret xyz987. The Consumer API executes the OAuth workflow and obtains the Access Token hij555 and the Token Secret lsn776. Using the HMAC-SHA1 signature method, the concatenated string xyz987&lsn776is used as the key to obtain the OAuth signature abd873=.

Here is an example of a request that returns information about posts:

```
GET http://api/quad/rest/posts
Host: api.quad.cisco.com:80
Authorization: OAuth realm="http://api.quad.cisco.com/api/quad/rest/posts",
    oauth_consumer_key="asd987",
    oauth_token="aln987",
    oauth_nonce="ain825",
    oauth_timestamp="1191242096",
    oauth_signature_method="HMAC-SHA1",
    oauth_version="1.0",
    oauth_signature="alkj@2983glkj@j"
```

Using xAuth for Trusted Clients

The xAuth protocol provides trusted clients and applications access to Cisco WebEx Social data via the Cisco WebEx Social API. An application or client is trusted when it maintains your Cisco WebEx Social login credentials (username and password). xAuth uses these credentials obtain an Access Token directly. The Get Request Token and the Get User Authorization steps that OAuth requires are not needed. With xAuth, a request is signed only with a Consumer Secret (with Oauth, a request is signed with both a Consumer Secret and a Token Secre)t.

When using xAuth, Cisco recommends that an API consumer use SSL to access Cisco WebEx Social. This approach ensures that Cisco WebEx Social login credentials are not exposed during a request transmission.

To use xAuth with trusted clients, you execute the Get xAuth Access Token and Access Secret operation, in which you provide the expired Access Token and related information.

Request

HTTP Method	URI
POST	**http://***server*[**:***port_number*]**/quadopen/oauth/xauth_access_token** where: • *server*—Host name or IP address of the Cisco WebEx Social server. • *port_number*—Cisco WebEx Social server port number that is used for communication with the API consumer. Must be provided if the port number is not 80.

Request Parameters

A Get xAuth Access Token and Access Secret request includes the following request parameters:

Request Parameter	Description
oauth_consumer_key	Consumer Key that was received as described in the "Authorization Flow Step 1: Add an API Consumer" section on page 2-3.
oauth_nonce	Unique random string that the API consumer provides to allow the Cisco WebEx Social server to verify that the request has not yet been made.

Request Parameter	Description
oauth_signature	Signature that verifies the request. Use the Consumer Secret to generate this signature. (The Consumer Secret was provided as described in the "Authorization Flow Step 1: Add an API Consumer" section on page 2-3.) If you are using the PLAINTEXT signature method, add an encoded & (%26) at the end of this string. If you are using HMAC-SHA1, add an unencoded & at the end of this string.
oauth_signature_method	Signing algorithm that is used generate the signature. Must be either **PLAINTEXT** or **HMAC-SHA1**.
oauth_timestamp	Timestamp that indicates when this request is submitted. This value must be specified in Unix time, which is the number of seconds since January 1, 1970 00:00:00 GMT, excluding leap seconds.
oauth_version	OAuth version being used. Must be **1.0**.
x_auth_username[1]	Cisco WebEx Social login username of the user that the client is obtaining a token on behalf of
x_auth_password[1]	Cisco WebEx Social login password of the user that the client is obtaining a token on behalf of
x_auth_mode[1]	Authorization mode. Must be **client_auth**.

1. This parameter must be included either in the request body or as a query parameter in the URI. It cannot be included in the HTTP header.)

The request parameters can be placed in any of these ways:

- Parameters except x_auth_username, x_auth_password, and x_auth_mode can be included in the authorization header of the request. In this case, use **Authorization: OAuth** in the header.
- As the body of the request. In this case, use **Content-Type: application/x-www-form-urlencoded** in the header.
- As query parameters in a request. The first query parameter must be preceded by a question mark (?). Separate each query parameter with an ampersand (&).

The following example shows a Get xAuth Access Token and Access Secret request that uses query parameters:

```
POST http://api.quad.cisco.com/quadopen/oauth/xauth_access_token
?x_auth_username=admin
&x_auth_password=passwordtest
&x_auth_mode=client_auth
&oauth_consumer_key=aoi1794g9713987
&oauth_nonce=mbeoih987234nlkeoi099
&oauth_signature=aj07%saldkj3nlkn%flkenagie16
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1203374975
&oauth_version=1.0
```

Response

The response to a Get xAuth Access Token and Access Secret request includes the following response parameters:

Response Parameter	Description
oauth_token	Access Token, which provides the API consumer with access to Cisco WebEx Social data
oauth_token_secret	Secret associated with the Access Token
oauth_session_handle	Credential used to refresh an expired valid Access Token
xoauth_user_id	Cisco WebEx Social User identifier that is associated with this Access Token

The following is an example of the response to a Get xAuth Access Token and Access Secret request:

```
oauth_token=kjhw83akjKJBE098mlklm09uNW&
oauth_token_secret=jaonqzc&
oauth_session_handle=dshj987qNLKNa3987nskSszn987LKJvlk&
xoauth_user_id=190206987
```