Configure Cisco Jabber

Configure Cisco Jabber

Cisco Jabber is a suite of Unified Communications applications that allow seamless interaction with your contacts from anywhere. Cisco Jabber offers IM, presence, audio and video calling, voicemail, and conferencing.

The applications in the Cisco Jabber family of products are:

- Cisco Jabber for Windows
- Cisco Jabber for Mac
- Cisco Jabber for iPhone and iPad
- Cisco Jabber for Android
- Cisco Jabber Softphone for VDI

For more information about the Cisco Jabber suite of products, see https://www.cisco.com/go/jabber or https://www.cisco.com/c/en/us/products/unified-communications/jabber-softphone-for-vdi/index.html .

For detailed information about how to configure your system to work with Cisco Jabber, see the *Cisco Jabber Deployment and Installation Guide* at http://www.cisco.com/c/en/us/support/unified-communications/jabber-windows/products-installation-guides-list.html.

OAuth Refresh Logins for Cisco Jabber

Cisco Jabber clients, as of Jabber Release 11.9, can use OAuth Refresh Logins to authenticate with Cisco Unified Communications Manager and the IM and Presence Service. This feature improves the user experience for Cisco Jabber by providing the following benefits:

- After an initial login, provides seamless access to resources over the life of the refresh token.
- Removes the need for Cisco Jabber clients to re-authenticate frequently.

- Provides consistent login behavior in SSO and non-SSO environments.

With OAuth Refresh Logins, Cisco Unified Communications Manager issues clusterwide access tokens and refresh tokens that use the OAuth standard. Cisco Unified Communications Manager and IM and Presence Service use the short-lived access tokens to authenticate Jabber (the default lifespan for an access token is 60 minutes). The longer-lived refresh tokens provide Jabber with new access tokens as the old access tokens expire. So long as the refresh token is valid the Jabber client can obtain new access tokens dynamically without the user having to re-enter credentials (the default refresh token lifespan is 60 days).

All access tokens are encrypted, signed, and self-contained using the JWT format (RFC7519). Refresh tokens are signed, but are not encrypted.

Note OAuth authentication is also supported by Cisco Expressway and Cisco Unified Connection. Make sure to check with those products for compatible versions. Refer to Cisco Jabber documentation for details on Jabber behavior if you are running incompatible versions.

Authentication Process

When a Cisco Jabber client authenticates, or when a refresh token is sent, Cisco Unified Communications Manager checks the following conditions, each of which must be met for authentication to succeed.

- Verifies the signature.
- Decrypts and verifies the token.
- Verifies that the user is an active user. For example, an LDAP-synced user whom is subsequently removed from the external LDAP directory, will remain in the database, but will appear as an inactive user in the User Status of End User Configuration.
- Verifies that the user has access to resources, as provided by their role, access control group, and user rank configuration.

Note For backward compatibility, older Jabber clients and supporting applications such as the Cisco Unified Real-Time Monitoring Tool can authenticate using the implicit grant flow model, which is enabled by default.

Cisco Jabber Prerequisites

The following prerequisites exist for Cisco Jabber integration:

- If you want to use OAuth Refresh Logins, you must enable the feature on all of your UC systems. Make sure that your Cisco Jabber, Cisco Unity Connection and Cisco Expressway deployments support OAuth refresh logins.
- If you are deploying Push Notifications for Cisco Jabber on iPhone or iPad, refer to Push Notifications for *Cisco Jabber on iPhone and iPad with Cisco Unified Communications Manager* for a complete list of Push Notifications prerequisites and configurations.

Cisco Jabber Configuration Task Flow

Complete these tasks in Cisco Unified Communications Manager to configure the system for Cisco Jabber clients.

Procedure

	Command or Action	Purpose
Step 1	Configure Refresh Logins for Cisco Jabber, on page 4	Enable Cisco Unified Communications Manager and the IM and Presence Service to use OAuth refresh logins for Cisco Jabber authentication. **Note** OAuth Refresh Logins are disabled by default in Cisco Unified Communications Manager, but are disabled by default in Cisco Expressway. If you choose not to enable the feature in Cisco Unified Communications Manager, you must disable the feature in Cisco Expressway or a configuration mismatch will result.
Step 2	Configure Push Notifications	If you are deploying Cisco Jabber for iPhone or iPad, enable Push Notifications on your system. **Note** Push Notifications is a mandatory configuration for Cisco Jabber on iPhone and iPad. The feature is not required for Android, Mac, or Windows users.
Step 3	Configure additional Cisco Jabber settings.	Refer to the *On-Premises Deployment for Cisco Jabber* guide for your platform. • Android—http://www.cisco.com/c/en/us/support/unified-communications/jabber-android/products-installation-guides-list.html • iPhone or iPad—http://www.cisco.com/c/en/us/support/customer-collaboration/jabber-iphone-ipad/products-installation-guides-list.html • Mac—http://www.cisco.com/c/en/us/support/unified-communications/jabber-mac/products-installation-guides-list.html

	Command or Action	Purpose
		• Windows—http://www.cisco.com/c/en/us/support/unified-communications/jabber-windows/products-installation-guides-list.html

Configure Refresh Logins for Cisco Jabber

Use this procedure to enable Refresh Logins with OAuth access tokens and refresh tokens in Unified Communications Manager. OAuth Refresh Logins provides a streamlined login flow that doesn't require users to re-login after network changes.

Note To ensure compatibility, make sure that the various Unified Communications components of your deployment, such as Cisco Jabber, Cisco Expressway and Cisco Unity Connection, support refresh logins. Once OAuth Refresh Logins are enabled, disabling the feature will require you to reset all Cisco Jabber clients.

Before you begin

You must be running a minimum release of Cisco Jabber 11.9. Older versions of Jabber will use the Implicit Grant Flow authentication model from previous releases.

Procedure

Step 1 From Cisco Unified CM Administration, choose **System** > **Enterprise Parameters**.

Step 2 Under **SSO Configuration**, do either of the following:

- Choose the **OAuth with Refresh Login Flow** enterprise parameter to **Enabled** to enable OAuth Refresh Logins.
- Choose the **OAuth with Refresh Login Flow** enterprise parameter to **Disabled** to disable OAuth Refresh Logins. This is the default setting.

Step 3 If you enabled OAuth Refresh Logins, configure expiry timers for access tokens and refresh tokens by configuring the following enterprise parameters:

- **OAuth Access Token Expiry Timer (minutes)**—This parameter specifies the expiry timer, in minutes, for individual OAuth access tokens. The OAuth access token is invalid after the timer expires, but the Jabber client can request and obtain new access tokens without the user having to re-authenticate so long as the refresh token is valid. The valid range is from 1 - 1440 minutes with a default of 60 minutes.
- **OAuth Refresh Token Expiry Timer (days)**—This parameter specifies the expiry timer, in days, for OAuth refresh tokens. After the timer expires, the refresh token becomes invalid and the Jabber client must re-authenticate to get a new refresh token. The valid range is from 1 - 365 days with a default of 60 days.

Step 4 Click **Save**.

Note Once you've saved the configuration, reset all Cisco Jabber and Webex clients.

Regenerate Keys for OAuth Refresh Logins

Use this procedure to regenerate both the encryption key and the signing key using the Command Line Interface. Complete this task only if the encryption key or signing key that Cisco Jabber uses for OAuth authentication with Unified Communications Manager has been compromised. The signing key is asymmetric and RSA-based whereas the encryption key is a symmetric key.

After you complete this task, the current access and refresh tokens that use these keys become invalid.

We recommend that you complete this task during off-hours to minimize the impact to end users.

The encryption key can be regenerated only via the CLI below, but you can also use the Cisco Unified OS Administration GUI of the publisher to regenerate the signing key. Choose **Security** > **Certificate Management**, select the **AUTHZ** certificate, and click **Regenerate**.

Procedure

Step 1 From the Unified Communications Manager publisher node, log in to the **Command Line** Interface .

Step 2 If you want to regenerate the encryption key:

a) Run the `set key regen authz encryption` command.
b) Enter `yes`.

Step 3 If you want to regenerate the signing key:

a) Run the `set key regen authz signing` command.
b) Enter `yes`.
The Unified Communications Manager publisher node regenerates keys and replicates the new keys to all Unified Communications Manager cluster nodes, including any local IM and Presence Service nodes.

You must regenerate and sync your new keys on all of your UC clusters:

- IM and Presence central cluster—If you have an IM and Presence centralized deployment, your IM and Presence nodes are running on a separate cluster from your telephony. In this case, repeat this procedure on the Unified Communications Manager publisher node of the IM and Presence Service central cluster.
- Cisco Expressway or Cisco Unity Connection—Regenerate the keys on those clusters as well. See your Cisco Expressway and Cisco Unity Connection documentation for details.

Note Restart the Cisco CallManager Service on all nodes in the cluster after the keys are reassigned.

Revoke Existing OAuth Refresh Tokens

Use an AXL API to revoke existing OAuth refresh tokens. For example, if an employee leaves your company, you can use this API to revoke that employee's current refresh token so that they cannot obtain new access tokens and will no longer be able to log in to the company account. The API is a REST-based API that is protected by AXL credentials. You can use any command-line tool to invoke the API. The following command provides an example of a cURL command that can be used to revoke a refresh token:

```
curl -k -u "admin:password" https://<UCMaddress:8443/ssosp/token/revoke?user_id=<end_user>
```

where:

- `admin:password` is the login ID and password for the Cisco Unified Communications Manager administrator account.
- `UCMaddress` is the FQDN or IP address of the Cisco Unified Communications Manger publisher node.
- `end_user` is the user ID for the user for whom you want to revoke refresh tokens.

Cisco Jabber Interactions and Restrictions

Feature	Interactions
Graceful registration	Graceful registration covers dual registration attempts from two Cisco Jabber clients with the same device name (for example, Jabber running on both an office laptop and a home office laptop). The feature de-registers the initial registration automatically so that the second registration can proceed. The de-registered Jabber client does not re-register. Graceful registration is supported automatically for Cisco Jabber, except when Jabber is deployed in a Mobile and Remote Access (MRA) deployment. In MRA deployments, the de-registered Jabber client attempts to re-register. For MRA deployments, if you have Cisco Jabber running on two devices with the same device name, make sure to log Jabber out of one device before you use the other.

Troubleshooting OAuth SSO Configuration

The following table highlights useful logs for troubleshooting OAuth SSO configuration. Trace does not need to be configured for these logs.

Note To set SAML SSO logs to a detailed level, run the `set samltrace level debug` CLI command.

Table 1: Logs for Troubleshooting OAuth Refresh Logins

Logs	Log Details
SSO Logs	Each time a new SSO App operation is completed, new log entries are generated here: `/var/log/active/platform/log/ssoApp.log`

Logs	Log Details
Ssosp Logs	SSO and OAuth operations are logged in ssosp logs. Each time SSO is enabled a new log file is created here: `/usr/local/thirdparty/Jakarta-tomcat/logs/ssosp/log4j/`
SSO and OAuth Configuration	Certificate logs are located at the following location. Each time the Authz certificate is regenerated, a new log file is generated: `/var/log/active/platform/log/certMgmt*.log`