BlueJeans

Relay Listener Service Installation Guide

Introduction

The Relay Listener Service is the client on-premise service that awaits commands from the Relay Cloud. These commands include endpoint controls (such as dial and disconnect), status inquiries (whether an endpoint is currently in a call), and Exchange Calendar polling (if enabled). It is supplied either as a Java .jar file or as RPM or DEB installer files that include a Java Service Wrapper.

Prerequisites

The Listener Service requires a Java Virtual Machine (JVM) of at least version 7. It has been tested with OpenJDK 7, and we recommend installation of the **openjdk-7-jre-headless** package on Ubuntu, or equivalent.

Network Requirements

The Listener Service will make an outbound connection to the AMQP host specified in the **config.properties** file via SSL on port **5671**. The Listener Service makes no other connections out to the Internet. It will perform DNS resolution if the host is given in FQDN format, which will require port TCP/UDP on port **53**.

Connections into your own internal network to control endpoints may be made on ports **22** (ssh), **80** (http) or **443** (https). The port used depends on the API mechanism for each type of endpoint (e.g., Lifesize, Polycom, Tandberg).

Installation or Upgrade

When using the RPM file, a listenerservice user and group will be created during installation, and init scripts will be put in place to start and stop the Listener Service much like any other service. Simply **rpm -Uvf listenerservice*.rpm** or **dpkg -i listenerservice*.deb** (Debian or Ubuntu) and then **service listenerservice start**.

To upgrade an existing installation, back up **config.properties** and **private.der** from the installation directory, then run the above RPM or DPKG installation command with the new installer.

When using the Listener Service from the command line from the .jar file, simply execute **java -jar listenerservice-jar-with-dependencies.jar**. You can also supply your own init.d scripts together with **start-stop-daemon** to run this command.

Configuration

Listener Service looks for a file called **config.properties**. Configuration values for this file are supplied by Blue Jeans and the file must be in the same directory as the listenerservice-jar-with-dependencies.jar file. It contains the AMQP host (see "Network Requirements" above), the username to use when logging in to that host, an encrypted form of the password to be used, and a unique listenerServiceId. See the comments in the file for examples and details.

We strongly recommend using Public Key Encryption (PKE) for storage of endpoint passwords provisioned with our cloud service. This requires generating an RSA key in DER format and then provisioning your RSA key with our cloud service. To generate a random RSA key, you can run the genssl.sh script from the Listener Service install directory, or manually using openssl:

```
openssl genrsa -out private.pem 2048
openssl pkcs8 -topk8 -in private.pem -outform DER -out private.der -
nocrypt
openssl rsa -in private.pem -pubout -outform DER -out public.der
```

Once you have your public.der file, it must be provisioned with your enterprise on our cloud service. You can do this yourself through the administration front-end or the REST API. The key must first be base64 encoded if you created the key manually. For example:

```
base64 < public.der
```

If you used the genssl.sh script, you will already have a base64-encoded version of your public key called public.b64.

Be sure that your private.pem and private.der files are readable only by the user that will run the Listener Service (e.g., chmod 600 private.*). It is also a best practice to keep a copy of your private.pem file in a safe place! If you lose your private key, it will not be possible to recover, decrypt, or re-encrypt any of your provisioned endpoint passwords; they will all have to be provisioned again.

The location of your private key file must also be added to config.properties:

```
encryption.private_key_file=/opt/listenerservice/private.der
```

If you opt not to use public key encryption, passwords are stored in an encrypted format that is difficult, but not impossible to decrypt. Using public key encryption ensures that only your own hosts with a copy of your private key on your own premises are capable of decrypting your endpoint passwords.

If you have existing endpoints with passwords provisioned without public key encryption, and then you provision a public key, the endpoints' encrypted passwords will be converted to use public key encryption *only when the endpoints are updated*. Any update will suffice, even an empty PATCH. This ensures that if there are any errors or problems with encryption setup, all endpoint passwords won't be immediately re-encrypted and potentially lost. It may be advantageous to update a single endpoint first and ensure that everything is working as expected before updating all endpoints.

Configuring the Listener Service for public key encryption does not preclude the default secret-key encrypted passwords from working; the two may coexist. We recommend, however, updating existing endpoints to use public key encryption as soon as practicable after provisioning the public key and verifying functionality.

See the provisioning guide and API documentation for more details about using the REST API.

Logs

Logging messages are saved to the **listenerService.log** file it a directory called **logs** off the directory containing the Listener Service .jar file. The log file will rotate when it reaches 1MB in size, and up to 10 old log files will be retained. When the Listener Service is run from the command line, logging messages also appear on the console.

Provisioning of endpoints and configuration of calendar polling are performed on the cloud service, not on the listener service; please see the Getting Started guide for these details.