Client Certificate Authentication for Vibes APIs

(Note that this page applies only when customers are calling the Vibes Message API and Public API systems. Customers who wish to ensure that callback requests from Vibes to their own systems are definitely being made by Vibes should consult Client Certificate Authentication for Callbacks from the Vibes Platform instead.)

The primary authentication mechanism for Vibes APIs (both the Message API system and the Public API system) is basic authentication using an email address as the user name. TLS client certificates may be used as an optional second authentication mechanism for customers who require it. (This is sometimes referred to as "mutual TLS authentication," since it is in addition to the authentication of the server that is always done as part of the TLS handshake.)

This requires configuration by both Vibes and the customer, as described below.

1. For each Vibes platform user ID that will be used to make API requests, the customer must generate and provide to Vibes a *certificate signing request* (CSR). Each CSR should have a subject distinguished name that includes the user ID that will be used with the certificate, and the name of the customer's company. The distinguished name in the CSR may also contain any other information desired by the customer.
2. Vibes will issue a TLS certificate for each CSR provided by the customer.
3. The customer must configure their system(s) that make API calls to Vibes systems so that they have access to the TLS certificate(s) and corresponding private key(s), and will be able to use those certificate(s) and key(s) during the TLS handshake when making API calls to VIbes systems.

Important Considerations

- As soon as Vibes configures the Message API and Public API systems to require client certificates for the customer account, *no API calls will be accepted by those systems without a client certificate.* This configuration setting applies to the entire customer account (but only to that specific account, not to any sub-accounts).
- The only certificate authority that is accepted by the Vibes API systems is the internal Vibes CA, and therefore *only the certificates issued by Vibes* in step 2 above may be used. No certificates issued by public CAs may be used for the purpose of authenticating to the Vibes API systems. This is necessary to maximize security for all API users who use client certificate authentication.

Examples of TLS Key and Certificate Generation

There are innumerable ways to generate TLS keys and certificates. This page presents sample commands that can be used to do this using a few common approaches.

Using Java keytool

If using a Java client system to make Vibes API calls, the most common way to generate TLS keys and certificates is using the keytool program provided as part of the Java Runtime Environment. Depending on the version of Java being used, keytool can be used to manipulate JKS ("Java Keystore") and/or PKCS#12 files. This example uses the JKS format.

Generate Private Key

First, a private key entry must be generated. JKS files may contain any number of entries, as long as each one has a different identifier or "alias." This sample command uses the alias "vibesclient1" to identify the key.

```
keytool -keystore example.jks -alias vibesclient1 -genkey -keyalg RSA -keysize 2048
```

This will prompt for a keystore password, first and last name, organizational unit, organization, city, state, and country code. These should be provided as appropriate. As mentioned above, *the "first and last name" field should contain the email address that will be used as a user ID when making API calls with this certificate,* and*the "organization" field should identify your company.* After all fields have been entered correctly, the keytool program will produce a keystore file (in this case "example.jks") or add an additional key entry if the keystore file already exists. You must remember the keystore password; it will be used every time the keystore is accessed.

Export CSR from keystore

Now that the private key has been generated, a CSR can be produced.

```
keytool -keystore example.jks -alias vibesclient1 -certreq -file vibesclient1.csr
```

This will produce a certificate signing request in a file called "vibesclient1.csr". If you inspect that file, you will see that it is a base-64 encoded string enclosed in "BEGIN NEW CERTIFICATE REQUEST" and "END NEW CERTIFICATE REQUEST" lines. That file must be provided to Vibes.

Import certificate into keystore

After Vibes generates a certificate from the CSR, we will return it, along with the Vibes CA Certificate, to you. These must both be imported into the keystore. When importing your certificate, you must use the same alias that you used before. The Vibes CA Certificate may be imported with any other alias that is not yet being used in the keystore file.

```
keytool -keystore example.jks -alias vibesca -import -file vibes-cacert.crt
```

This command will prompt "Trust this certificate?"; you will need to respond "yes" to proceed.

```
keytool -keystore example.jks -alias vibesclient1 -import -file vibesclient1.crt
```

This command should produce the output "Certificate was added to keystore". That indicates that the command was successful.

The resulting keystore file, along with the keystore password, can be provided to the client programs that will be submitting requests to Vibes APIs.

Using the openssl command-line program

OpenSSL is an open source program that implements the TLS protocol. Among other things, it can be used to produce TLS keys and certificates. This process is slightly more straightforward than using the Java keytool program.

To generate a private key and CSR, a single command can be used:

```
openssl req -outform PEM -out vibesclient1.csr -newkey rsa:2048 -keyout vibesclient1.key -sha256 -
verbose
```

This will prompt for a PEM pass phrase (with confirmation), country name, state name, locality, organization name, organizational unit, common name, and possibly other fields like email address. As mentioned above, *the "first and last name" field should contain the email address that will be used as a user ID when making API calls with this certificate,* and*the "organization" field should identify your company.* Other fields may be specified however seems most appropriate. You must remember the pass phrase; it will be needed whenever using the private key.

The program will produce two files, a private key (in this example, "vibesclient1.key") and a CSR ("vibesclient1.csr"). The private key file must be kept private, and the CSR must be provided to Vibes.

Vibes will generate a certificate from the CSR and will return it, along with the Vibes CA Certificate, to you. When using OpenSSL, unlike the Java keytool program, the Vibes CA certificate is not actually needed. The certificate and private key files, along with the PEM pass phrase, must be provided to client programs that will submit requests to Vibes APIs.