

# KeyTalk automated Apache certificate renewal installation instructions

## Introduction

---

Serving HTTPS traffic via Linux-based Apache webserver requires assigning a server SSL certificate for each SSL virtual host defined by Apache. Typically, these certificates have a long lifetime. Decreasing certificate lifetime improves security, but increases maintenance work to generate and assign new certificates. This document explains you how to setup the KeyTalk Linux client to automatically renew Apache SSL certificates.

When set up, KeyTalk will periodically perform the following actions for each configured Apache SSL virtual host:

1. Check the lifetime of the currently installed certificate
2. If the certificate is about to expire (typically less than 10% lifetime):
  - a. Retrieve a new SSL certificate from your KeyTalk server
  - b. Install and effectuate the retrieved certificate to the configured Apache SSL virtual host

The remaining document describes how to configure your KeyTalk client and server for automatic Apache SSL virtual host certificate renewal.

## System Requirements

---

Before installing the KeyTalk client with Apache certificate renewal script please make sure that your server meets the following requirements:

- Ubuntu 20.04 LTS x64 or Ubuntu 22.04 LTS x64 or Ubuntu 24.04 LTS x64
- Apache 2.2 - 2.4 installed with SSL module enabled

Example configuring Apache:

```
sudo apache2ctl -v if ! sudo a2query -m ssl; then
sudo a2enmod ssl;
sudo a2ensite default-ssl;
sudo service apache2 restart;
fi
```

## Compatible KeyTalk versions

---

This admin manual requires KeyTalk client v7.8.0 or higher and KeyTalk server v7.8.7 and higher

## KeyTalk server configuration

---

Before running the script, a KeyTalk service definition and appropriate users for server certificate distribution must be configured:

1. Log into your KeyTalk server web interface.
2. Configure a service that will be used to provide the Apache SSL certificates
3. Enable `serverAuth` in the `Extended Key Usage` section for the selected service
4. Bind authentication module backend bound to the selected service
5. Create a user for the selected authentication module You may choose to use your SSL domain name (e.g. `example.com`) for user name. This user name will be appear as X.509 `Common Name` in the generated SSL certificate
6. Configure `Subject Alternative Name` either on per-service or on per-user basis. Notice that most modern browsers check `Subject Alternative Name` certificate extension when matching SSL domain name, hence partially or totally ignoring the value written in the `Common Name`

## KeyTalk client configuration

---

### Install and customize KeyTalk client

*Before installing* the KeyTalk client, make sure you have Apache installed, otherwise the Apache certificate renewal script will not be installed.

Extract:

```
tar -xf KeyTalkClient-<version>.linux.tar
```

Install using:

```
sudo ./install.sh
```

Alternatively, you can install from .deb package:

```
sudo apt -y update
sudo apt -y install KeyTalkClient_<version>.deb
```

Customize KeyTalk installation with a RCCD file:

```
sudo /usr/local/bin/keytalk/ktconfig --rccd-path <path-to-rccd>
```

Notice: RCCD customization for Apache integration must be done as root because the Apache certificate renewal script invoked by cron is also run as root and therefore will expect the KeyTalk configuration under '/root/.keytalk'

Test your installation is correct:

```
sudo /usr/local/bin/keytalk/ktclient --provider <keytalk-provider> --service <keytalk-service> --user <keytalk-user>
> --password <keytalk-password>
```

## Configure Apache certificate renewal

Update the Apache certificate renewal configuration found at `/etc/keytalk/apache.ini` :

1: Specify Apache virtual host(s) for which you want KeyTalk to automatically renew SSL certificates

Example IP-based VHost:

```
"VHost" : "*:443"
```

Example name-based VHost:

```
"VHost" : "*:443",
"ServerName" : "www.example.com",
```

2: Specify KeyTalk server credentials

Example passwordless login:

```
"KeyTalkUser" : "www.example.com",
```

Example login with password:

```
"KeyTalkUser" : "www.example.com",
"KeyTalkPassword" : "secret",
```

3: Configure email notifications

Example through local mailserver:

```
"EmailNotifications" : true,
"EmailFrom" : "me@example.com",
"EmailTo" : "you@example.com",
```

Example fully customized notifications:

```
"EmailNotifications" : true,
"EmailFrom" : "me@example.com",
"EmailTo" : "you@example.com",
"EmailServer" : "smtp.example.com"
"EmailSubject" : "Apache certificate update"
```

Do a test run:

```
sudo /usr/local/bin/keytalk/renew_apache_ssl_cert
```

In your browser, navigate to the virtual hosts you configured.

Once everything works as expected, enable certificate renewal cron job by uncommenting the relevant line in `/etc/cron.d/keytalk-apache`

Enjoy!

# Troubleshooting

## My current certificate won't renew

By default, the renewal script will only refresh the certificate if the currently installed certificate for this VHost has left than 10% of it's validity duration left. In some cases, such as first-time installation of a certificate, a changed `apache.ini` (VHost configuration), or changed KeyTalk service configuration, you may want to force a refresh sooner.

To force a refresh, even though the current certificate is not about to expire, use:

```
sudo /usr/local/bin/keytalk/renew_apache_ssl_cert --force
```

## I'm not receiving e-mails for a certain VHost

All VHosts for which you want to send e-mails should contain the following setting:

```
"EmailNotifications" : true
```

Some parameters of the E-mail configuration are optional. For example when no `EmailServer` is specified, then a correctly configured mail server running at `localhost` is assumed.

If you don't have a running mail server, you can specify an SMTP mail server such as:

```
"EmailServer" : "smtp.example.com"
```

## How can I temporarily disable renewal for a virtual host?

In `/etc/keytalk/apache.ini` you can comment out all lines for this VHost by prepending them with a hash sign `#`. After you have done this, make sure that the content which is not commented out still has the correct braces `{}` and commas.

For example if you want to disable the VHost on port 3001 in this configuration:

```
{
  "VHost" : "*:3000",
  "ServerName" : "a.example.com",
  "KeyTalkProvider" : "MyProvider",
  "KeyTalkService" : "MY_SERVICE",
  "KeyTalkUser" : "a.example.com"
}, {
  "VHost" : "*:3001",
  "ServerName" : "b.example.com",
  "KeyTalkProvider" : "MyProvider",
  "KeyTalkService" : "MY_SERVICE",
  "KeyTalkUser" : "b.example.com"
}
```

You can comment out all lines for the VHost on port 3001 by prepending all lines with `#`:

```
{
  "VHost" : "*:3000",
  "ServerName" : "a.example.com",
  "KeyTalkProvider" : "MyProvider",
  "KeyTalkService" : "MY_SERVICE",
  "KeyTalkUser" : "a.example.com"
# }, {
# "VHost" : "*:3001",
# "ServerName" : "b.example.com",
# "KeyTalkProvider" : "MyProvider",
# "KeyTalkService" : "MY_SERVICE",
# "KeyTalkUser" : "b.example.com"
}
```

Now only the host on port 3000 is considered and the lines starting with # will be treated as if they were blank.

## My certificate is not trusted by browsers at the client.

Possible reasons are:

- The browser at your client did not install the required CA certificates to verify the full certificate chain.
- There is a `Subject Alternative Name` defined in the browser which takes precedence over the Common name. If this is the case, please update (or remove) the `Subject Alternative Name` field in your KeyTalk service.
- Your apache virtual host configuration has an incorrect `ServerName` directive.

## I still can't solve my problem

Please create a problem report with

```
sudo /usr/local/bin/keytalk/ktprgen
```

and send it to support@keytalk.com