# Installation and Configuration

December 16, 2012

This document describes how to install the Virtual Safe application on a new server.

## 1 Software needed

The Virtual Safe application needs the following software to be installed. For reference, we also give the version we used during development, as well as the corresponding FreeBSD port development was made on a FreeBSD jail available by all the members using git).

- Apache 2.2.23 (`www/apache22` port)
- OpenSSL 1.0.1 (`security/openssl`)
- MySQL 5.5.28 (`databases/mysql55-server`)
- PHP 5.4.9 (`lang/php5`), with the following plugins
  - Mcrypt (`security/php5-mcrypt`)
  - PDO for MySQL (`databases/php5-pdo_mysql`)

The next sections will describe the configuration of those softwares.

## 2 Web Service

### 2.1 SSL

First of all, the server's certificate should be created:

```
mkdir /etc/ssl/{private,cert}
chmod 0700 /etc/ssl/{private,cert}
openssl req -x509 -nodes -days 365 -out /etc/ssl/cert/server.crt -keyout /etc/ssl/private/
chmod 0400 /etc/ssl/private/server.key /etc/ssl/cert/server.crt
```

## 2.2 Apache

The following virtual host and directory entry should be added to Apache's configuration, where `/home/secu/www/` is the directory that contains the sources of the web service. The following listing could be for example be saved in `conf/extra/httpd-vhosts.conf` in the configuration directory of Apache, and the line `Include conf/extra/httpd-vhosts.conf` should then be added to Apache's main configuration file (`httpd.conf`).

```
NameVirtualHost *:443
<VirtualHost *:443>
     ServerName cours.awesom.eu
     # Only the public part of the site is allowed
     DocumentRoot /home/secu/www/public
     SSLEngine on
     SSLCertificateFile /etc/ssl/cert/server.crt
     SSLCertificateKeyFile /etc/ssl/private/server.key
</VirtualHost>

<Directory /home/secu/www/public>
  Order allow,deny
  Allow from all
  DirectoryIndex index.php
  # Turn off all option to improve security
  Options None
  # Disable .ht* files
  AllowOverride None
</Directory>
```

### 2.2.1 Strengthening the default Apache configuration

The following configuration lines are optional, but should be done on the production server to ensure maximal security. They can be put in Apache's main configuration file (`httpd.conf`).

```
# Don't show Apache's version
ServerSignature Off
ServerTokens Prod

# Decrease the timeout to mitigate DDOS attacks
Timeout 50
```

## 2.3 PHP

PHP should be activated in Apache, so the following line should be added to Apache's main configuration file:

```
AddType application/x-httpd-php .php
```

The modules used by the web service should also be activated in PHP's `php.ini`:

```
extension=mcrypt.so
extension=openssl.so
extension=pdo_mysql.so
```

# 3 Database

After MySQL have been installed, we have to create the users and tables needed. Change the passwords of the MySQL in the file `sql/users.sql`. Then, load the files in `.sql` in the following way:

```
# cd sql/
# mysql -u root -p
> source database.sql
> source users.sql
> source schema.sql
> source test_data.sql # optional, add some users and admin
```

## 3.1 Admin creation

It is not possible to add an administrator from the web application. The 'admin' database user should do it directly in SQL. A convenience script that creates the line to add is available in `sql/new_user.sh` and take the admin name as argument.

### 3.1.1 Strengthening the default MySQL configuration

The script `mysql_secure_installation` provided with MySQL can be used to strengthen the default configuration. Among the things it does, it changes the root password, removes anonymous users, disallow remote root login, remove the test databases, ...

# 4 Directories

The following directories should also be created, and be owned by the web user (the user with which Apache is launched, `http` or `www` on most systems). The `data` directory should not be readable by other users:

```
# mkdir -p data/{certificates,pubkeys,files}
# chown -R www data/ # if apache is launched by the www user
# chmod -R o-rwx data/
```

# 5 Java Application

The Java application requires Bouncy Castle[1] to be able to read keys formatted with the PEM format. On most Linux distribution, it is available in the package `bcprov`.

For convenience, a `.jar` containing bouncy castle and the Java application is provided in the directory `signer/`. Thus, for launching the Java application, one should just do:

```
java -jar signer.jar [params]
```

The application is only usable in command line (but is made in such a way that it is easy to add a graphical user interface). The web service always give the parameters to pass to the application.

[1] `http://www.bouncycastle.org/java.html`