

Installation Guide

Virtual Security Lab

by Alexis SEMREN and Steven MARIEN

History	Who	Version	When
Creation	AGGS3/SM861	1.0	19/08/2015

1 - Introduction	1
2 - Process	1
2 - 1 Apache2/PHP5	1
2 - 2 Composer	1
2 - 3 Database	1
2 - 5 Initialisation	2
2 - 6 User	2
3 - Check	2
4 - To go further	3
4 - 1 Virtual host	3
4 - 2 Alternative PHP Cache (APC)	3

1 - Introduction

This guide will assume you are using a Debian 7. This guide requires to be able to use apt-get install and to have an Internet connection. The project source can be found at this address https://kentlab.github.io/virtual_security/Source_Code/source.zip

The following command line **supposes you are in the root of the application folder.**

2 - Process

2 - 1 Apache2/PHP5

One of the first things to perform is to install apache2 and php5 with the following command. This command will install the web server and php5.

```
apt-get update  
apt-get install apache2 php5 php5-intl
```

Then, download the source of the project and place it in this folder **/var/www/**.

2 - 2 Composer

Once the first step is done, go in the project folder, then download **composer** and install it, according to the documentation <https://getcomposer.org/download/>. Curl might be a dependency. If you don't have curl, you can still use the php command line. Once the **composer.phar** is downloaded, type:

```
php composer.phar install
```

This command can take time depending on the machine and connection, as it will download all the necessary dependencies for the project.

Sometimes, when git is missing a problem can happen. It can be fixed by installing it. Another problem can be authenticating to Github, to resolve it just add **—prefer-source** in the composer command line.

2 - 3 Database

The installation is almost done. Just install the **sqlite3** which is a Relational Database Management Systems (RDMS) used in this project following this command.

```
apt-get install sqlite3 php5-sqlite
```

Regarding the database, there is no configuration to perform because sqlite3 database is a file already included in the project source (**database/lab.db**).

2 - 5 Initialisation

In order to make things easier, a script is available in the source (**deploy.sh**). This script will do the necessary, like remove the cache, make a copy of the database, etc...

```
sh deploy.sh  
or  
./deploy.sh
```

2 - 6 User

In order to use the admin part, a user with an admin role is required. Basically, the user admin is **KentLabAdmin** with the password **KentF0rever**. The project is using the bundle named **FosUserBundle**, so the users can be managed according to the documentation http://symfony.com/doc/current/bundles/FOSUserBundle/command_line_tools.html

Disable user

```
php app/console fos:user:deactivate KentLabAdmin
```

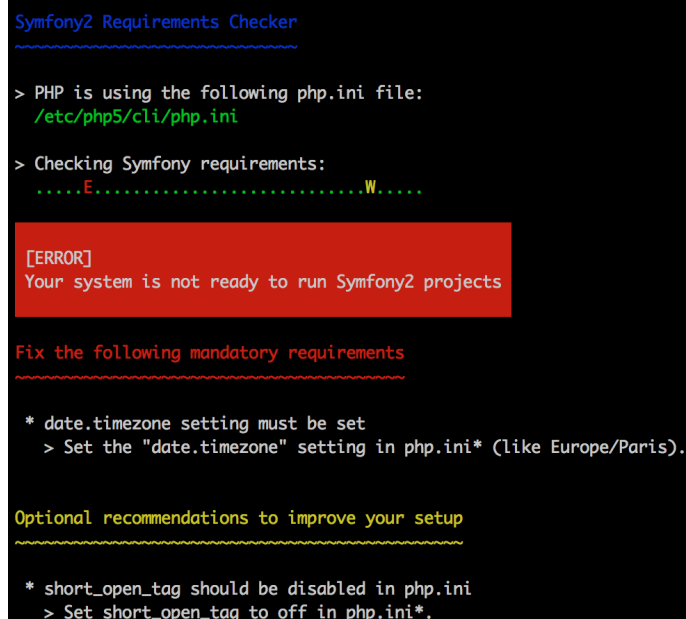
Create user

```
php app/console fos:user:create testuser test@example.com p@ssword --super-admin
```

3 - Check

At this point, you should be able to use the website without any problem following the url **{ip}/lab/web/app.php**. You can run the following command to find out if you have any problem. If you have a problem, this file will tell you how to fix it. When you get warnings, it is just to advise you to have a more performant application.

```
php app/check.php
```



```
Symfony2 Requirements Checker  
~~~~~  
> PHP is using the following php.ini file:  
  /etc/php5/cli/php.ini  
> Checking Symfony requirements:  
  ....E.....W.....  
  
[ERROR]  
Your system is not ready to run Symfony2 projects  
  
Fix the following mandatory requirements  
~~~~~  
* date.timezone setting must be set  
  > Set the "date.timezone" setting in php.ini* (like Europe/Paris).  
  
Optional recommendations to improve your setup  
~~~~~  
* short_open_tag should be disabled in php.ini  
  > Set short_open_tag to off in php.ini*.
```

4 - To go further

4 - 1 Virtual host

Basically, the urls given by Symfony2 are not really pretty. According to the [Symfony2 documentation](http://symfony.com/doc/current/cookbook/configuration/web_server_configuration.html), a virtual host can be added. The documentation is available on http://symfony.com/doc/current/cookbook/configuration/web_server_configuration.html. Don't forget to activate the url rewriting.

```
a2enmod rewrite
service apache2 restart
```

Here is an example of a virtual host which may be added in **/etc/apache2/site-enabled/000-default:**

```
<VirtualHost *:80>
...

DocumentRoot /var/www/lab/web
<Directory /var/www/lab/web>
    AllowOverride All
    Order Allow,Deny
    Allow from All
</Directory>

...
</VirtualHost>
```

You can now access to the application through this url **{ip}/** instead of **{ip}/lab/web/app.php**.

4 - 2 Alternative PHP Cache (APC)

In order to increase the website speed, it can be interesting to install the APC which is a cache system for php. To do it, it is necessary to install the following package.

```
apt-get install php-apc
```

Then, add this line at the end of **php.ini** file edit before.

```
[APC]
extension = apc.so
apc.enabled = 1
```