

# **EIP Vigilate**

*Installation documentation 2 (ID2)*

15 décembre 2016

## *Résumé du document*

Cette partie explique comment installer les 3 parties de la solution Vigilate dans un environnement de production. Les environnements de production et de développement sont quasiment identiques pour le backend et le scanner de programme, car ils sont codés en python et ont juste besoin d'un éditeur de texte pour être développés. Le frontend, quant à lui, nécessite des outils supplémentaires pour l'environnement de développement. Pour le backend, il faut essentiellement python, django, nginx, un gestionnaire de bases de données, quelques modules python, et uwsgi qui fera la liaison entre nginx et Django. Pour le scanner, juste python3 et le module requests. En ce qui concerne le frontend, il va falloir installer nodejs et npm, un gestionnaire de paquets pour Javascript. Linux est indispensable pour la partie serveur. Il faut ensuite vérifier que Python 3 est à jour. Le serveur est en Django, il faut donc l'installer ainsi que les autres modules python nécessaires. Pour finir l'installation, il faut ensuite cloner le projet depuis github. Une série de commandes décrite plus bas, est utile pour le frontend en nginx. Pour installer les dépendances vers le scanner de programme, il y a 3 méthodes différentes selon l'OS : mac, linux ou windows.

### *Description du document*

|                          |   |
|--------------------------|---|
| <b>Titre</b>             | [2017][Vigilate][ID2]                       |
| <b>Date</b>              | 15/12/2016                                  |
| <b>Auteur</b>            | Kévin SOULES                                |
| <b>Responsable</b>       | Kévin SOULES                                |
| <b>Email</b>             | vigilate_2017@labeip.epitech.eu             |
| <b>Sujet</b>             | Installation documentation 2                |
| <b>Mots clés</b>         | ID2, sécurité, vulnérabilités, architecture |
| <b>Version du modèle</b> | 1.1   |

### *Tableau des révisions*

| <b>Date</b> | <b>Auteur</b>  | <b>Section(s)</b>            | <b>Commentaires</b>                             |
|-------------|----------------|------------------------------|---|
| 30/04/16    | Daniel Mercier | Documentation d'installation | Rédaction de la documentation d'installation    |
| 01/05/16    | Kevin Soules   | Toutes                       | Création du document latex / formatage          |
| 01/05/16    | Prune Budowski | Résumé                       | Rédaction du résumé                             |
| 05/07/16    | Daniel Mercier | Documentation d'installation | Amélioration de la documentation d'installation |
| 06/07/16    | Prune Budowski | Résumé                       | Amélioration du résumé                          |
| 07/07/16    | Kevin Soules   | Toutes                       | Création du document latex / formatage          |

# Table des matières

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Prérequis</b>                                  | <b>1</b> |
| 1.1      | Prérequis pour le Backend . . . . .               | 1        |
| 1.2      | Prérequis pour le scanner de programmes . . . . . | 2        |
| 1.3      | Prérequis pour le frontend . . . . .              | 2        |
| <b>2</b> | <b>Environnement de production</b>                | <b>3</b> |
| 2.1      | Partie Serveur . . . . .                          | 3        |
| 2.2      | Scanner de programmes . . . . .                   | 7        |
| 2.3      | Frontend . . . . .                                | 8        |
| <b>3</b> | <b>Environnement de développement</b>             | <b>9</b> |
| 3.1      | Backend . . . . .                                 | 9        |
| 3.2      | Scanner de programmes . . . . .                   | 9        |
| 3.3      | Frontend . . . . .                                | 9        |

# Chapitre 1

## Prérequis

Vigilate à besoin d'un certain nombre de prérequis logiciel pour fonctionner. Dans les parties suivantes, il est expliqué comment installer ces différents prérequis. Chaque composant à ces propres prérequis.

### 1.1. Prérequis pour le Backend

- Une installation de linux (l'explication est donnée pour ubuntu)
- python3
- pip
- git
- la version developpeur de python3
- un compilateur
- mysql et/ou postgresql
- nginx
- uwsgi

Le backend à aussi besoin des modules python3 suivant :

- django version 1.9
- django-filter version 0.11.0
- djangorestframework version 3.2.4
- docutils version 0.12
- Markdown version 2.6.2
- MarkupSafe version 0.23
- mysqlclient version 1.3.6
- packaging version 15.3
- pycpg2 version 2.6.1
- Pygments version 2.0.2
- PyMySQL version 0.6.7
- python-dateutil version 2.4.2
- pytz version 2015.7

- requests version 2.8.1
- six version 1.9.0
- snowballstemmer version 1.2.0
- Sphinx version 1.3.1
- sphinx-rtd-theme version 0.1.9
- SQLAlchemy version 1.0.8
- SQLAlchemy-Utills version 0.31.0
- virtualenv version 13.1.2
- Werkzeug version 0.10.4
- django-nose version 1.4.3
- coverage version 4.1
- coveralls version 1.1
- django-phonenumbers-field version 1.1.0
- django-bulk-update version 1.1.10
- stripe version 1.37.0
- django-ratelimit version 1.0.0
- raven version 5.31.0

## **1.2. Prérequis pour le scanner de programmes**

- git
- python3
- le module python3 requests

## **1.3. Prérequis pour le frontend**

- git
- npm
- nodejs

## Chapitre 2

# Environnement de production

### 2.1. Partie Serveur

L'installation de la partie serveur de Vigilate est basée sur plusieurs scripts utilisés par Travis pour exécuter nos tests d'intégration continue. Pour commencer l'installation, une installation de Linux est nécessaire, l'explication est faite pour une installation sur la distribution Ubuntu mais est facilement adaptable pour n'importe quelle distribution Linux. Généralement remplacez

```
apt-get install
```

par la commande d'installation de paquets de votre distribution.

Avant toute installation mettez à jour vos dépôts avec la commande

```
user $ sudo apt-get update
```

Un compilateur est nécessaire pour certaines des installations qui suivent, pour en installer un utilisez la commande :

```
user $ sudo apt-get install python3-dev build-essential binutils
```

Python 3 est nécessaire pour la partie serveur de Vigilate, il est installé par défaut sur Ubuntu et bon nombre d'autres distributions. Si jamais ce n'est pas le cas, utilisez la commande

```
user $ sudo apt-get install python3 python-setuptools
```

La partie serveur de vigilate est stockée sur un serveur git, il faut donc installer git afin de pouvoir la récupérer. Pour cela utilisez la commande

```
user $ apt-get install git
```

Afin d'installer django, il faut utiliser le gestionnaire de paquet python (pip). Pour l'installer utilisez la commande :

```
user $ sudo easy_install pip3
```

Pour récupérer la partie serveur de Vigilate, utilisez les commandes suivantes :

```
user $ git clone https://github.com/vigilate/backend.git
```

Puis allez dans le dossier nouvellement créé avec la commande suivant.

```
user $ cd backend
```

Le backend de Vigilate peut utiliser 2 différents gestionnaires de bases de données, mysql ou postgresql. Vous pouvez installer l'un des 2 ou les 2. Pour installer mysql utilisez :

```
user $ sudo apt-get install mysql-server libmysqlclient-dev
```

Pour installer postgresql utilisez la commande :

```
user $ sudo apt-get install postgresql postgresql-server-dev-9.4
```

Le backend de vigilate à aussi besoin de uwsgi, pour l'installer utilisez :

```
user $ sudo pip3 install uwsgi
```

La partie serveur de Vigilate est basée sur Django, pour installer Django et les autres modules python nécessaire utilisez la commande

```
user $ sudo pip3 install -r requirements.txt
```

Pour le stockage des mots de passe Vigilate utilise un hash appelé Argon2, il faut l'installer avec la commande suivante :

```
user $ sudo ./install-pyargon2.sh
```

Avant de créer la base de données de Vigilate il faut créer un user sur mysql, voici les commandes à utiliser :

```
user $ mysql -u root -p
```

(un mot de passe sera demandé, mettez toor).

```
user $ mysql> CREATE USER 'vigilate'@'%' IDENTIFIED BY 'superpwd';
```

(remplacer superpwd par votre mot de passe pour cette commande et les suivantes).

```
user $ mysql> GRANT ALL PRIVILEGES ON vigilate.* TO vigilate@'%' IDENTIFIED BY 'superpwd';
```

```
user $ mysql> GRANT ALL PRIVILEGES ON test_vigilate.* TO vigilate@'%' IDENTIFIED BY 'superpwd';
```

```
user $ mysql> CREATE DATABASE vigilate;
```

Maintenant, il faut créer la base de données de Vigilate, pour cela exécutez les commandes suivantes.



```
user $ python3 manage.py makemigrations vigilate_backend
```

```
user $ python3 manage.py makemigrations vulnerability_manager
```

```
user $ python3 manage.py migrate
```

```
user $ python3 manage.py migrate --run-syncdb
```

Il faut ensuite mettre des valeurs par défaut en base de données via la commande suivante :

```
user $ python3 manage.py loaddata offers
```

Ensuite il faut créer un superuser, pour ça utilisez la commande suivante : (répondez aux question)

```
user $ python3 manage.py createsuperuser
```

Dans le fichier backend/settings.py il faut trouver la section DATABASE. Pour la database 'default' (qui est MySQL) remplacer 'root' par vigilate et 'toor' par le mot de passe utiliser lors de la création d'utilisateur dans MySQL.

Dans le fichier /etc/postgresql/9.4/main/pg\_hba.conf. il faut changer "local all postgres peer" par "local all postgres trust".

Afin de vérifier que toutes la configuration est correct lancez les test via la commandes :

```
user $ python3 ./manage.py test
```

Maintenant il faut configurer uwsgi, utilisez la commande suivante :

```
user $ echo H4sIAAqxglcAAzWPQW7CMBBF93OKXCD4BF5QNUJRpUqtQSxQFsYZ\
3BETOxrbUN8eC9rtN683758OgfIE75ic0JopBl2OZjd2w7KiRIHt\
JaPoVBNHv8lWPGaAk0G5kcMJh190psVZq5JEcXSW1ZmCKvfkqet7\
CtSpn7igupEnthnV2borhvlV2bQCfGN6MizfbU3wQcyGfLCszbj7\
Oox72NcVdYiZLhXavzBbmQdp9q8Pm8bZ3DlBqHm+QYGpV5gqMN\
Gee3qpfCmfqSUP6nPABolsVqAAEAAA== | base64 -d | gzip -d |\
sudo tee /lib/systemd/system/uwsgi.service
```

Une fois cette commande exécutée, il faut éditer le fichier de configuration, pour cela ouvrez le fichier `/lib/systemd/system/uwsgi.service` avec votre éditeur de texte favori est

remplacé chaque occurrence de `/home/vigilate` par le chemin vers le dépôt du backend que vous avez cloné précédemment.

Il faut aussi chaque occurrence de 'vigilate' par votre user courant dans le fichier `uwsgi.ini`.

Puis activez l'exécution au démarrage du service :

```
user $ sudo systemctl enable uwsgi
```

Le frontend de Django est nginx, pour l'installer utilisez la commande

```
user $ sudo apt-get install nginx
```

Pour configurer nginx il faut utiliser les commandes suivante :

```
user $ echo H4sIAN2hg1cAA41STW/bMAw9z79CtwCFBAUYihbLyUiMpVjTBIuL\
7WaoEmNzk6XMkvPRIv+9jO2uw4AC4UEiqPfIR4rtNsQGVM3ML+VK\
z14SRhag2UHDWoeHL1LKytcgdl1iVRHkk9K/wRkZPN1xkpySZMC/\
JB3bYoJgWGe340kX05VqAsQu1saNuJ30WG0RXCxqdSievDkWAZ+B\
3Vwvela7DyUWpM8UEWvwbWSfx+OBar1WEb1jsgaDig3az6YsqsA\
0N2h+/yn/zOFsI6+NFOP/iDVP0n6NrYqBDbMefL3DZ22rYH01xC1\
dCW6g3yJNkoOQ7+DIGOKigZC4x6lWkMIYupdbLwVqbV+L5YNyXQj\
Nroakfq9OhL/0wW8BcTKm0DEr1nO2erxfKT5dM7ZLLvP8oyzeZbO\
OFuu8rvlw5qtluv8vcTF+ubd47nO7CHnP8W0DdHXfZR/A9gSDnfA\
H2mpRFRsDhDoO/xpgbbKiB8YK363EQtvcIMUWNMEgU+VruCtEj/f\
RBT5cQs8bamxBp+7f3kXPPzaKXkFGewJtWMDAAA= | base64 -d | gzip -d |\
sudo tee /etc/nginx/conf.d/vigilate.conf
```

Une fois cette commande exécutée, il faut éditer le fichier de configuration, pour cela ouvrez le fichier `/etc/nginx/conf.d/vigilate.conf` avec votre éditeur de texte favori est remplacé chaque occurrence de `/home/vigilate` par le chemin vers le dépôt du backend que vous avez cloné précédemment.

Il faut maintenant installer postfix afin de pouvoir envoyer des mails. Pour cela utilisez la commandes suivante :

```
user $ sudo apt-get install postfix
```

Redémarrez nginx via la commande :

```
user $ sudo service nginx restart
```

Maintenant, il faut initialiser les bases de données de vulnérabilités et de version de programmes.

Cette opération peut prendre près d'une heure pour cela utilisez les commandes suivantes :

```
user $ wget http://127.0.0.1/update_cpe
user $ wget http://127.0.0.1/update_all_cve
```

## 2.2. Scanner de programmes

Pour le scanner du programme, il faut avoir python 3 d'installé.

Si vous êtes sous Linux utilisez la commande

```
user $ sudo apt-get install python3
```

Pour Mac os X allez sur la page : <https://www.python.org/downloads/mac-osx/> et choisissez l'une des versions python 3 disponible et installez là.

Pour les utilisateurs Windows allez sur la page <https://www.python.org/downloads/windows/> et téléchargez l'un des installateurs python 3 et installez le.

Maintenant, il faut installer les dépendances du scanner de programme, pour cela, il faut commencer par installer le gestionnaire de paquets de python pip. Pour Linux ouvrez un terminal et lancez la commande

```
user $ sudo easy_install pip3
```

Pour Mac os X et Windows pip est supposé être installé avec python, si ce n'est pas le cas suivez les instructions sur cette page : <https://pip.pypa.io/en/stable/installing/> Pour installer les dépendances sous Linux ou Mac os X utilisez la commande :

```
user $ sudo pip3 install requests
```

Pour windows ouvrez un terminal et utilisez les commandes suivantes :

Pour windows 7 :

```
user $ cd AppData\Local\Programs\Python\Python35
```

Pour Windows 10 :

```
user $ cd AppData\Local\Programs\Python\Python35-32
```

Ensuite, la commande est la même pour les 2 versions de Windows :

```
user $ python.exe Scripts\pip.exe install requests
```

Maintenant, il faut récupérer le scanner de programmes il y a 2 solutions, en utilisant git (voir l'installation de la partie serveur pour installer git) via la commande (pour Linux ou Mac os)

```
user $ git clone https://github.com/vigilate/program_scanner.git
```

Pour Windows cloner le dépôt [https://github.com/vigilate/program\\_scanner.git](https://github.com/vigilate/program_scanner.git) Pour une installation plus simple, vous pouvez télécharger une version zippé du scanner de programme avec ce lien : [https://github.com/vigilate/program\\_scanner/archive/master.zip](https://github.com/vigilate/program_scanner/archive/master.zip).

Dézippez le et utilisez la commande suivante pour exécuter le scanner de programme : Windows 7 :

```
user $ AppData\Local\Programs\Python\Python35\python.exe chemin\scanner.py
```

Windows 10 :

```
user $ AppData\Local\Programs\Python\Python35-32\python.exe scanner.py
```

Linux :

```
user $ python3 chemin/vers/le/dossier/du/scanner/scanner.py
```

Mac os X :

```
user $ python3 chemin/vers/le/dossier/du/scanner/scanner.py
```

### 2.3. Frontend

Le frontend angular2 est du code statique qui est hébergé sur les serveur web de github (github pages). Il n'y a donc rien à installer concernant l'environnement de production. Il suffit juste de copier la version compilée (cf 3.3 environnement de développement) des fichiers javascript vers le dépôt git <https://github.com/vigilate/vigilate.github.io> et il sera automatiquement disponible à l'adresse <https://vigilate.github.io>

## Chapitre 3

# Environnement de développement

### 3.1. Backend

Le backend de Vigilate est codé entièrement en python, le seul outil nécessaire pour le développer est donc un éditeur de texte. Ce qui veut dire que l'installation de l'environnement de développement est identique à celle de l'environnement de production.

### 3.2. Scanner de programmes

Le scanner de programmes est lui aussi codé en python, un éditeur de texte est donc suffisant pour son développement. Cependant afin de pouvoir le tester et le développer sur tous les systèmes d'exploitation compatible, avoir des machines fonctionnant sous les différent système est une bonne pratique. Ces machines peuvent être des machines virtuel. Les OS compatibles sont Mac os, Windows, Linux basé sur le gestionnaire de paquets pacman, Linux basé sur le gestionnaire de paquets sur apt-get, Linux basé sur le gestionnaire de paquets sur rpm et BSD (qui utilise pkg comme gestionnaire de paquets).

### 3.3. Frontend

Commencez par installer git (cf 2.1 Partie Serveur), puis faites les commandes suivantes pour installer node :

```
user $ curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
```

```
user $ sudo apt-get install -y nodejs
```

Une fois fais, vous pouvez installer angular-cli :

```
user $ sudo npm install -g angular-cli
```

Clonez le dépo frontend sur votre machine :

```
user $ git clone https://github.com/vigilate/frontend
```

Installez les dépendances :

```
user $ cd frontend
```

```
user $ npm install
```

Vous êtes maintenant prêt pour développer. Pour compiler le projet, vous pouvez utiliser la commande suivante :

```
user $ ng build
```

Pour servir les fichiers, vous pouvez aussi lancer

```
user $ ng serve
```