

Программа для ЭВМ Vul-Search

Руководство по установке и настройке

Оглавление

1. Общие сведения.....	2
1.1. Назначение.....	2
1.2. Системные требования.....	2
2. Обзор процесса установки.....	2
3. Описание процесса установки.....	2
3.1. Запуск системы в контейнере.....	2
3.2. Ручной запуск.....	3
3.2.1. Подготовка окружения.....	3
3.2.2. API.....	3
3.2.3. Frontend.....	4
3.3. Локальный запуск для разработки.....	4
3.3.1. Скрипты краулеров.....	4
3.3.2. Скрипт импорта.....	4
3.3.3. Ошибки при импорте.....	5

1. Общие сведения

1.1. Назначение

Программный продукт – сервис Vul-Search предназначен для автоматизированного сбора, хранения и обработки публично доступной информации об уязвимостях программного обеспечения. А также для выявления известных уязвимостей в заданных наименованиях программного обеспечения методом сопоставления наименования программного обеспечения с информацией, хранящейся в базе знаний об уязвимостях.

1.2. Системные требования

Минимальные требования к оборудованию для работы Сервиса:

- Процессор Intel Ice Lake, 2 CPU;
- 4 ГБ ОЗУ;
- Диск объемом 500 ГБ.

Также для установки необходимо подключение к сети Интернет.

2. Обзор процесса установки

Сервис поставляется в виде zip-архива Vul-Search.zip, содержащего набор файлов исходного кода на языке JavaScript, а также файлов метаданных, предназначенных для сборки docker-образа, для запуска сервиса в виде контейнера.

3. Описание процесса установки

3.1. Запуск системы в контейнере

Запуск происходит с помощью docker-compose, для запуска нужно выполнить команду:

```
docker compose up -d
```

При первом запуске автоматически стартуют скрипты краулеров, которые собирают данные из источников уязвимостей и импортируют их в базу данных. После завершения скриптов система будет готова к работе.

Доступные переменные окружения:

- API_HOST - на каком хосте запускать api (по умолчанию: 0.0.0.0);
- API_PORT - на каком порте запускать api (по умолчанию: 4000);
- API_ENABLE_CORS - нужно ли обрабатывать cors, 1 - да 0 - нет, нужно для локальной работы (по умолчанию: 0);
- API_INDEX_NAME - название индекса откуда читать в elasticsearch (по умолчанию: v13s);
- API_ELASTICSEARCH_HOST - хост elasticsearch (по умолчанию: http://es:9200).

3.2. Ручной запуск

3.2.1. Подготовка окружения

- Необходимо установить elasticsearch

```
docker run -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" -e "http.cors.enabled=true" -e "http.cors.allow-origin=*" -e "http.cors.allow-headers=X-Requested-With,X-Auth-Token,Content-Type,Content-Length,Authorization" -e "http.cors.allow-credentials=true" docker.elastic.co/elasticsearch/elasticsearch:7.15.0
```

- Необходимо установить nodejs

```
curl -sL https://deb.nodesource.com/setup_17.x | bash -
```

- Настройка nginx - нужно настроить отдачу статики и проксирование на api сайта по пути /api

3.2.2. API

- Заходим в папку - cd api
- Ставим зависимости - npm install
- Запускаем скрипт -

```
HOST=127.0.0.1 PORT=4000 node index.js
```

Параметры запуска:

- API_HOST - на каком хосте запускать api (по умолчанию: 127.0.0.1)

- API_PORT - на каком порте запускать api (по умолчанию: 40000)
- API_ENABLE_CORS - нужно ли обрабатывать cors, 1 - да 0 - нет, нужно для локальной работы (по умолчанию: 0)
- API_INDEX_NAME - название индекса откуда читать в elasticsearch (по умолчанию: v13s)
- API_ELASTICSEARCH_HOST - хост elasticsearch (по умолчанию: http://localhost:9200)

3.2.3. Frontend

- Заходим в папку - `cd /`
- Ставим зависимости - `npm install`
- Собираем статику - `REACT_APP_API_HOST=/api npm run build`
- Забираем готовую статику из папки - `build`

Параметры запуска:

- REACT_APP_API_HOST - адрес api сайта (по умолчанию: http://localhost:4000)

3.3. Локальный запуск для разработки

- Заходим в папку - `cd /`
- Ставим зависимости - `npm install`
- Запускаем api сайта с включенным cors - `API_ENABLE_CORS=1 node index.js`
- Запускаем dev сервер статики - `npm run start`
- По адресу localhost:3000 запустить сервер со статикой

3.3.1. Скрипты краулеров

- Заходим в папку - `cd scripts/crawlers/cve`
- Ставим зависимости - `npm install`
- Запускаем скрипт -

`node index.js`