

Руководство разработчика «Рутокен VPN Community Edition»

Версия 1.0

Основные сведения	4
Термины, определения и сокращения	5
Ссылки на репозитории	6
Быстрый старт	6
Руководство пользователя и администратора	6
Руководство пользователя	6
Руководство администратора	6
Интернационализация	6
Деплой сервера	6
Зависимости проекта и их взаимодействие	7
Описание компонентов системы	7
Описание взаимодействия компонентов системы	8
Настройка центра сертификации	8
Структура папок проекта	9
Стек технологий	11
Документация frontend'а	12
Контроллеры Backend'а	13
Описание методов контроллеров	13
AuthController	13
IndexController	14
LogsController	14
PersonalController	14
SettingsController	14
StatusController	15
SystemController	15
UsersController	16
Модели (классы)	18
Backend	18
ADBBackend	18
ConfigForm	18
ConfigNetForm	18
ChangeAdminPass	19
LOGS_PATH	19

ConfigSettings	19
ConfigNetwork	20
TaskStatus	20
ConfigVpn.....	21
ConfigPki	21
ConfigNtp	21
ConfigRouting.....	22
ConfigLogs	22
AccessToCertGeneration.....	22
Frontend.....	22
IdentityModel	22
LoginModel	23

Основные сведения

Рутокен VPN Community Edition (далее "Рутокен VPN CE") представляет собой клиент-серверное решение, предназначенное для обеспечения удобного и безопасного доступа к инфраструктуре компании. Продукт основан на решении Рутокен VPN, разработанном компанией "Актив", которое, в свою очередь, базируется на программном продукте OpenVPN. OpenVPN реализует технологию VPN для создания зашифрованных каналов.

Подробная описание располагается в [README.md](#) файле.

Термины, определения и сокращения

CA — Certification Authority (Центр сертификации)

CRL — Certificate Revocation List (Списки отозванных сертификатов)

OpenVPN — свободная реализация технологии виртуальной частной сети (VPN) с открытым исходным кодом для создания зашифрованных каналов типа точка-точка или сервер-клиенты между компьютерами

OpenSSL — криптографическая библиотека с открытым исходным кодом, широко известна из-за расширения SSL/TLS, используемого в веб-протоколе HTTPS

SSL — криптографический протокол, который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений

TLS — Transport Layer Security, Протокол защиты транспортного уровня, как и его предшественник SSL - криптографические протоколы, обеспечивающие защищённую передачу данных между узлами в сети Интернет

UI — User Interface, пользовательский интерфейс

VPN — Virtual Private Network (виртуальная частная сеть)

ИКТ — Информационно-коммуникационные технологии

ПК — персональный компьютер

ПО — программное обеспечение

Рутокен VPN CE — Рутокен VPN Community Edition

ФСБ — Федеральная служба безопасности Российской Федерации

ФСТЭК — Федеральная служба по техническому и экспортному контролю

Центр сертификации OpenSSL — Модуль библиотеки OpenSSL, позволяющий создать полноценную инфраструктуру открытых ключей

ЦС — Центр сертификации

Ссылки на репозитории

Репозиторий проекта - <https://github.com/AktivCo/Rutoken-VPN-Community-Edition-Server>

Быстрый старт

Руководство для быстрого запуска РутOKEN VPN CE на машине разработчика описано в файле [DEVELOPERS.md](#)

Описание для быстрого запуска РутOKEN VPN CE на сервере вы можете найти в файле [INSTALL.md](#)

Руководство пользователя и администратора

Руководство пользователя

С руководством пользователя можно ознакомиться по [ссылке](#)

Руководство администратора

С руководством администратора можно ознакомиться по [ссылке](#)

Интернационализация

РутOKEN VPN CE не поддерживает интернационализацию интерфейса.

Деплой сервера

Руководство для деплоя сервера РутOKEN VPN CE вы можете найти в файле [DEPLOY.md](#)

Зависимости проекта и их взаимодействие

Описание компонентов системы

Система РУТОКЕН VPN сервер использует четыре основных компонента:

- [OpenVPN](#)
- [OpenSSL](#)
- Центр сертификации OpenSSL
Модуль библиотеки OpenSSL , позволяющий создать полноценную инфраструктуру открытых ключей
- Рутокен VPN CE сервер

Взаимодействие компонентов представлено на рис. 1.

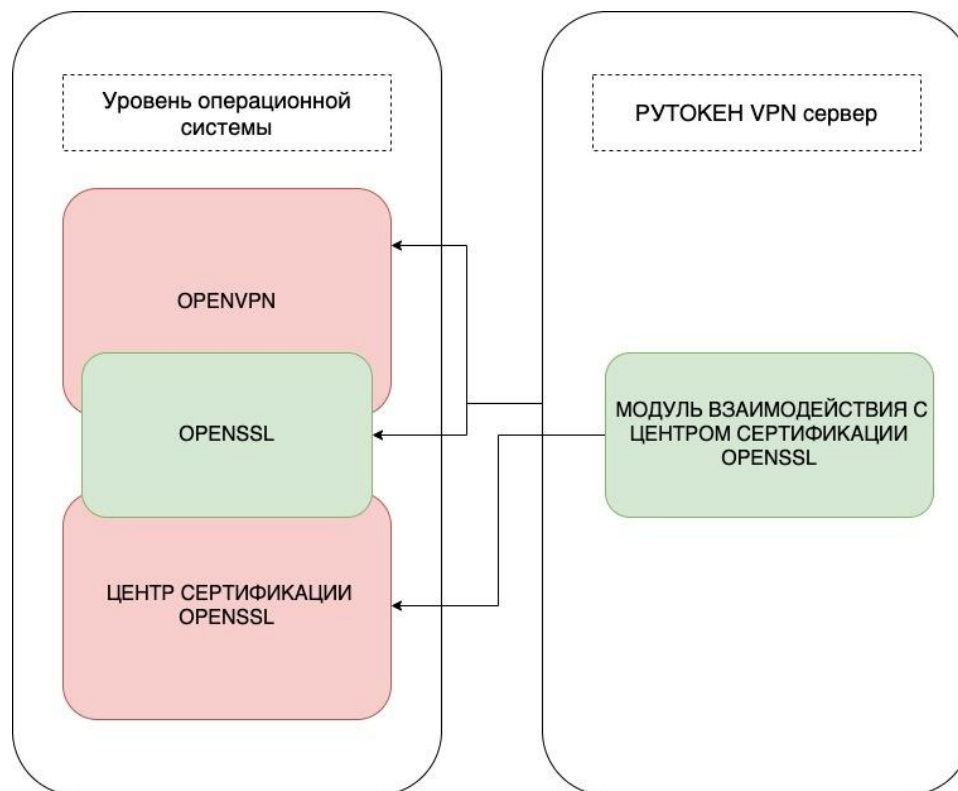


Рисунок 1. Взаимодействие компонентов системы

Описание взаимодействия компонентов системы

1. OpenVPN - OpenSSL.

OpenSSL является основой для аутентификации по сертификатам OpenVPN и механизмом создания зашифрованных соединений. На основе инфраструктуры PKI OpenSSL организуется механизм аутентификации пользователей.

2. OpenSSL - центр сертификации OpenSSL.

Центр сертификации — это модуль библиотеки OpenSSL, позволяющий создать полноценную инфраструктуру открытых ключей. В рамках настройки создается корневой ключ и сертификат, ключ и сертификат VPN сервера, изготавливаются сертификаты пользователей. OpenSSL обеспечивает аутентификацию по протоколу TLS.

3. Модуль взаимодействия с центром сертификации - центр сертификации OpenSSL.

В приложении реализован модуль, который позволяет выполнять вызовы к библиотеке OpenSSL, настраивать центр сертификации. С помощью модуля возможно выполнить полный цикл настройки центра сертификации.

Настройка центра сертификации

В основе работы системы находится центр сертификации. Реализован центр сертификации с помощью OpenSSL и модуля - обертки, отвечающего за управление центром со стороны UI.

Порядок настройки следующий:

- Генерация ключевой пары и сертификата центра сертификации;
- Генерация ключевой пары и сертификата vpn server - а;
- Генерация списка отзыва;







После первоначальной настройки доступны следующие операции:

- Выписывание пользовательского сертификата;
- Отзыв пользовательского сертификата;
- Ведение базы данных и формирования списка отзыва CRL;




Структура папок проекта

Файловая структура проекта:

vpn/









-  config/openssl.cnf - openssl конфиг
-  __init__.py - файл инициализации модуля
-  middleware.py - промежуточные обработчики http запросов
-  settings.py - основные конфигурации проекта
-  urls.py - файл сопоставления url запроса к обработчику в контроллере
-  wsgi.py - wsgi конфигурация

front/

-  app/ - директория с файлами фронтенда
-  environment/data-definitions/data-defenitions.opensource.ts - enum с наименованием приложения
-  styles/ css - стили приложения

 pkiapi/ - директория содержит набор методов для работы с PKI.

vpnserver/

-  static/ - директория в которую в последующем собирается фронтенд.
-  views/ - директория с обработчиками запросов
-  auth_back.py - методы для аутентификации пользователей, для взаимодействия с Active Directory
-  environment.py - метод для получения текущего environment приложения
-  helpers.py - методы для получения текущего статуса центра сертификации (настроен/ в процессе настройки/ не настроен)
-  logs_path.py - пути файлов логирования
-  models.py - модели (таблицы БД) проекта
-  os_helper.py - вспомогательные методы для осуществления операций с элементами операционной системы

- 📄 tasks.py - методы, описывающие действия по созданию инфраструктуры PKI
- 📄 users_helper.py - вспомогательные методы для работы с пользователями
- 📄 __index__.html - главная html страница.
- 📄 DEVELOPERS.md - Инструкция запуска сервиса управления на машине разработчика.
- 📄 INSTALL.md - Инструкция установки сервиса с помощью скрипта install.sh.
- 📄 manage.py - файл используется при выполнении install.sh, устанавливает переменные окружения.
- 📄 .eslintrc.js - конфигурационный файл lint
- 📄 .htmlhintrc - конфигурационный файл htmlhint
- 📄 .prettierrc.js - конфигурационный файл prettier
- 📄 angular.json - конфигурационный файл angular
- 📄 babel.config.js - конфигурационный файл babel
- 📄 setProduction.js - конфигурационный файл с константой определяющей тип сборки frontend'а проекта
- 📄 webpack.config.babel.js - конфигурационный файл webpack

Стек технологий

Backend:

- Python
- Django

Frontend:

- JavaScript
- TypeScript
- Webpack
- Angular 10
- RXJS 6
- npm
- node.js

Связанные продукты:

- [Рутокен VPN Клиент](#)
- [Openvpn](#)
- [Openssl](#)
- [Рутокен плагин](#)
- [Токены Рутокен](#)

Документация frontend'а

Документация для front-end части формируется с помощью утилиты для автоматической генерации документации:

[Compodoc](#)

Документация располагается в папке `documentation/`*

Формирование документации:

```
npm install
```

```
npm run compodoc
```

Контроллеры Backend'a

API работает по протоколу HTTP.

Данные на сервер отправляются при помощи POST методов в формате application/x-www-form-urlencoded.

Ответы возвращаются в формате JSON.

Пример запроса:

HTTP METHOD GET: /api/certinfo

Ответ:

```
JsonResponse {  
    needRequest : (boolean)  
    startDate: (date)  
    endDate: (date)  
    serial: (string)  
    fingerPrint: (string)  
}
```

Описание методов контроллеров

AuthController

Контроллер аутентификации

Методы:

signin(request) - Аутентификация пользователя

Параметры:

login (string): Логин пользователя

password(string): Пароль пользователя

signout(request) - Логаут пользователя

IndexController

Контроллер, возвращающий главную страницу.

Методы:

index(request) - Возвращает главную страницу

LogsController

Контроллер для работы с логами

Методы:

get_logs_list(request) - Возвращает логи системы

clear_logs(request) - Очищает логи

logs_enable(request) - Включает логирование

Параметры:

set_is_enable_to(boolean): Включение/отключение логирования

PersonalController

Контроллер для обработки пользовательских запросов

Методы:

vpn_getclientvpnconf(request) - Скачивание vpn-конфига

Параметры:

set_is_enable_to(boolean): Включение/отключение логирования

personal(request) - Выписывание сертификата пользователя

Параметры:

name(string): Имя пользователя

cert_req(string): PKCS10 запрос на сертификат

SettingsController

Контроллер для установки системных настроек

Методы:

vpn_config_admpwd(request) - Смена пароля администратора

Параметры:

password0(string): Пароль администратора

password1(boolean): Новый пароль администратора

manage_box(request) - Выключение/перезагрузка устройства

Параметры:

action(number): Тип операции перезагрузка/выключение

vpn_config_ntp(request) - Настройка поставщиков времени

StatusController

Контроллер для получения системных состояний

Методы:

identity(request) - Получение информации о текущем пользователе

init_status(request) - Состояние настройки центра сертификации

set_task_init(request) - Установка состояния центра сертификации на начальное значение

SystemController

Контроллер для настройки системы

Методы:

vpn_config_network(request) - Настройка сети

Параметры:

server_ip(string): IP адрес сервера

server_mask(string): Маска сервера

server_gate(string): Шлюз сервера

server_dns1(string): ДНС адрес сервера (1)

server_dns2(string): ДНС адрес сервера (2)

vpn_config_pki(request) - Настройка центра сертификации

Параметры:

common_name(string): Common Name корневого сертификата ЦС

pki_type(string): Тип центра сертификации

vpn_config_vpn(request) - Настройка vpn сервера

Параметры:

server_name(string): Имя VPN сервера

external_ip(string): IP для подключения клиентов к VPN серверу

vpn_config_domain(request) - Настройка домена

Параметры:

domain_server(string): Адрес доменного сервера

ldap_base_dn(string): DN доменного сервера

name(string): Имя пользователя домена с правами просмотра каталога

password(string): Пароль пользователя

vpn_config_routing(request) - Настройка маршрутизации

vpn_cert_info(request) - Информация о сертификате vpn сервера

UsersController

Контроллер для работы с пользователями

Методы:

get_domainusers_list(request) - Получение списка пользователей домена

users(request) - Получение списка пользователей системы

vpn_config_crl(request) - Отзыв сертификата

Параметры:

certificate(string): CommonName сертификата для отзыва

vpn_config_connected_users(request) - Получение списка подключенных пользователей

vpn_config_disconnect_user(request) - Отключение пользователя

Параметры:

client(string): Имя пользователя

sync_with_ad(request) - Синхронизация списка пользователей с Active Directory

Модели (классы)

Backend

ADBackend

Класс, использующийся для соединения с Active Directory

Методы:

`authenticate(self, username=None, password=None)` - Аутентифицирует пользователя по `username` и `password`. Возвращает аутентифицированного пользователя.

`get_user(self, user_id)` - Получает пользователя из Active Directory по `user_id`. Возвращает пользователя в случае его нахождения или `null`.

ConfigForm

Класс, генерирующий поля для формы настроек сервера

Свойства:

`common_name`: string - Название компании

`server_name`: string - Доменное имя сервера Rutoken VPN CE, или Ip адрес

`domain_server`: string - Адрес контроллера домена

`ldap_base_dn`: string - Основное имя домена

ConfigNetForm

Класс, генерирующий поля для формы сетевых настроек сервера

Свойства:

`MODE` - флаг, отвечающий за тип ip.

Значения:

`m` - Статический.

`a` - DHCP.

server_ip: string - IP адрес сервера
server_mask: string - Маска подсети
server_gate: string - Адрес шлюза подсети
server_dns1: string - DNS сервер 1
server_dns2: string - DNS сервер 2 (опционально)

ChangeAdminPass

Класс, генерирующий поля для формы смены пароля Администратора системы

Свойства:

password1: string - новый пароль Администратора
password2: string - подтверждения нового пароля Администратора

LOGS_PATH

Класс с константами путей до лог-файлов и директорий с лог-файлами

Свойства:

LOGS_DIR: string - путь до директории с логами
LOG_FILE_NAME: string - название лог-файла
LOG_FILE_PATH: string - путь до лог-файла
LOGROTATE_DIR: string - директория конфигурации для логирования
LOGRORATE_FILE_PATH: string - файл конфигурации для логирования

ConfigSettings

Класс, отвечающий за настройки сервера

Свойства:

TYPE - enum с типами шифрования. В Рутокен VPN CE доступно только rsa.
server_key: string - Ключ сервера OpenVpn
domain_server: string - Адрес контроллера домена

name: string - Пользователь для подключения к AD
password: string - Пароль пользователя
ldap_base_dn: string - Основное имя домена
server_type: string - выбранный тип шифрования

ConfigNetwork

Класс сетевых настроек сервера

Свойства:

MODE - enum с типами режимов работы.

Значения:

a - auto

m - manual

server_ip: string - Ip адрес сервера

server_mask: string - Маска подсети

server_gate: string - Адрес шлюза подсети

server_dns1: string - DNS сервера #1"

server_dns2: string - DNS сервера #2"

server_eth_mode: string - выбранный тип режима работы

TaskStatus

Класс, отвечающий за статусы операций приложения

Свойства:

STATUS_CHOICES - enum статусов операций

Значения:

INIT = 0

STARTED = 1

FINISHED = 2

TYPE_CHOICES - enum состояний системы: PKI_TYPE -происходит настройка ЦС, UPDATE_TYPE - обновляется система, REBOOT_TYPE - система перезагружается

Значения:

NONE_TYPE = 0

PKI_TYPE = 1

UPDATE_TYPE = 2

REBOOT_TYPE = 3

status: int - текущий статус. Возможные значения - STATUS_CHOICES. По умолчанию INIT.

type: int - текущий тип. Возможные значения - TYPE_CHOICES. По умолчанию NONE_TYPE.

description: string - описание статуса.

ConfigVpn

Класс VPN настроек

Свойства:

server_name: string - Внешний "белый" IP адрес сервера

cipher: string - Шифрование. Значение по умолчанию - "BF-CBC"

external_ip: string - Внешний "белый" IP адрес сервера

ConfigPki

Класс с данными для настроек PKI

Свойства:

common_name: string - Название компании. По умолчанию - RutokenVpn

pki_type: string - тип PKI. По умолчанию - rsa

ConfigNtp

Класс настроек NTP сервера

Свойства:

ntp_server: string - адрес NTP сервера

ConfigRouting

Класс настроек маршрутизации

Свойства:

ip: string - IP адрес

mask: string - маска сети

ConfigLogs

Класс настроек логирования

Свойства:

is_enabled: boolean - Включено/отключено логирование. По умолчанию выключены (false).

level: int - Текущий уровень логирования. По умолчанию уровень = 3.

AccessToCertGeneration

Класс, отвечающий за доступ пользователя к операциям генерации сертификатов

Свойства:

user - пользователь, который будет выполнять операции

can_generate_mobile_cert: boolean - Может ли пользователь генерировать мобильный сертификат. По умолчанию - true

can_generate_cert_on_token: boolean - Может ли пользователь генерировать сертификат на токене. По умолчанию - true

Frontend

IdentityModel

Класс, описывающий информацию и права залогиненного пользователя

Свойства:

id: number - уникальный идентификатор пользователя

username: string - имя пользователя

fullname: string - полное имя пользователя

isDomain: boolean - является ли пользователь доменным (Active Directory)

canGenerateMobileCert: boolean - может ли пользователь сгенерировать мобильный сертификат

canGenerateCertOnToken: boolean - может ли пользователь сгенерировать сертификат на токене

isDemoMode: boolean - находится ли пользователь в демо-режиме

LoginModel

Класс, описывающий модель пользователя, который логинится в систему

Свойства:

login: string - логин пользователя

password: string - пароль пользователя