### ****Финальная версия технического задания (ТЗ) для проекта VPN-сервера с Telegram-ботом****

### ****1. Цель проекта****

Создать инфраструктуру для предоставления бесплатных VPN-соединений через Telegram-бота. Система должна включать:

* Установку и настройку OpenVPN.
* Использование RSA ключей для шифрования.
* Базу данных для управления пользователями и трафиком.
* Генерацию конфигураций .ovpn.
* Telegram-бот для взаимодействия с пользователями. Проект предусматривает возможность масштабирования и добавления платных тарифов в будущем.

### ****2. Функциональные требования****

#### ****2.1 VPN-сервер****

* **Тарифы**:
  + Бесплатный тариф (10 ГБ/мес).
  + Возможность добавления платных тарифов в будущем.
  + Возможность создания спецтарифа (безлимит) для администраторов.
* **Функциональность**:
  + Генерация .ovpn конфигураций для каждого пользователя.
  + Автоматический перевод пользователей на бесплатный тариф при истечении лимита трафика.
* **Шифрование и безопасность**:
  + AES-256-CBC для шифрования.
  + SHA512 для аутентификации.
* **Сетевые параметры**:
  + VPN-сеть: 10.8.0.0/24.
  + Протокол: UDP, порт 1194.
  + IPv6: поддержка сети fddd:1194:1194:1194::/64.
* **Хранение файлов**:
  + Клиентские конфигурации: /etc/openvpn/client/.

#### ****2.2 Telegram-бот****

* **Функционал**:
  + Команды:
    - /start — приветственное сообщение.
    - /help — инструкция по настройке VPN для популярных ОС.
    - /getconfig — отправка конфигурационного файла.
    - /gettraffic — получение остатка трафика.
  + Логика:
    - Проверка существования .ovpn файла перед генерацией.
    - Генерация конфигурации через generate\_user\_config.sh.
    - Отправка файла через FSInputFile.
* **Сохранение данных пользователей**:
  + Telegram ID, имя пользователя, тариф, остаток трафика, статус оплаты, путь к конфигурации.
* **Уведомления**:
  + Информирование о превышении лимита трафика.
  + Напоминания о необходимости смены тарифа.
* **Файлы бота**:
  + Расположение: /opt/tg\_bot.

#### ****2.3 Безопасность****

* Установка OpenSSH.
* Открытие только необходимых портов:
  + 22 (SSH) — для администрирования.
  + 1194 (UDP) — для OpenVPN.
* Закрытие всех других портов через ufw.
* Автоматическая блокировка подозрительных IP через fail2ban.

#### ****2.4 База данных****

* Использование SQLite для хранения данных.
* Расположение: /opt/tg\_bot/base/vpn\_users.db.
* **Структура базы данных**:
  + Таблица users:

CREATE TABLE users (

id INTEGER PRIMARY KEY AUTOINCREMENT,

telegram\_id INTEGER UNIQUE,

username TEXT,

tariff TEXT,

traffic\_left INTEGER,

expiry\_date TEXT,

payment\_status TEXT,

config\_path TEXT

);

* + Таблица traffic:

CREATE TABLE traffic (

id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER,

date TEXT,

bytes\_used INTEGER,

FOREIGN KEY(user\_id) REFERENCES users(id)

);

### ****3. Структура файлов и каталогов****

1. **Каталог OpenVPN** (/etc/openvpn/):
   * server/server.conf — настройки OpenVPN.
   * easy-rsa/ — RSA ключи и сертификаты.
   * /etc/openvpn/client/ — клиентские конфигурации.
2. **Каталог Telegram-бота** (/opt/tg\_bot/):
   * bot.py — код Telegram-бота.
   * generate\_user\_config.sh — скрипт генерации .ovpn.
   * base/vpn\_users.db — база данных.
   * logs/ — логи работы бота.

### ****4. Технические требования****

#### ****4.1 Установка ПО****

1. **OpenSSH**:

sudo apt update && sudo apt install openssh-server -y

sudo ufw allow 22

sudo ufw enable

1. **OpenVPN**:

sudo apt update && sudo apt install openvpn easy-rsa -y

1. **SQLite**:

sudo apt install sqlite3 -y

1. **Fail2Ban**:

sudo apt install fail2ban -y

#### ****4.2 Настройка RSA ключей****

1. Инициализация PKI:

make-cadir /etc/openvpn/easy-rsa

cd /etc/openvpn/easy-rsa

./easyrsa init-pki

1. Создание ключей:
   * CA:

./easyrsa build-ca nopass

* + Сервер:

./easyrsa gen-req server nopass

./easyrsa sign-req server server

* + Клиенты:

./easyrsa build-client-full <client\_name> nopass

#### ****4.3 Настройка OpenVPN****

1. Создайте /etc/openvpn/server/server.conf:

local 45.76.38.87

port 1194

proto udp

dev tun

ca ca.crt

cert server.crt

key server.key

dh dh.pem

auth SHA512

tls-crypt tc.key

topology subnet

server 10.8.0.0 255.255.255.0

server-ipv6 fddd:1194:1194:1194::/64

push "redirect-gateway def1 ipv6 bypass-dhcp"

ifconfig-pool-persist ipp.txt

push "dhcp-option DNS 1.1.1.1"

push "dhcp-option DNS 1.0.0.1"

push "block-outside-dns"

keepalive 10 120

user nobody

group nogroup

persist-key

persist-tun

verb 3

crl-verify crl.pem

explicit-exit-notify

1. Запустите сервис:

sudo systemctl enable openvpn@server

sudo systemctl start openvpn@server

#### ****4.4 Настройка Telegram-бота****

1. Создайте бота через @BotFather и получите токен.
2. Настройте окружение:

cd /opt/tg\_bot

python3 -m venv venv

source venv/bin/activate

pip install aiogram sqlite3

1. Создайте bot.py (последний рабочий код бота).
2. Создайте generate\_user\_config.sh:

#!/bin/bash

CLIENT\_NAME=$1

OUTPUT\_DIR="/etc/openvpn/client/"

EASY\_RSA\_DIR="/etc/openvpn/easy-rsa"

cd $EASY\_RSA\_DIR

./easyrsa build-client-full $CLIENT\_NAME nopass

cp $EASY\_RSA\_DIR/pki/issued/$CLIENT\_NAME.crt $OUTPUT\_DIR/$CLIENT\_NAME.ovpn

1. Настройте сервис:

sudo nano /etc/systemd/system/vpn\_bot.service

[Unit]

Description=HideWay Bot VPN Service

After=network.target

[Service]

Type=simple

User=root

WorkingDirectory=/opt/tg\_bot

ExecStart=/opt/tg\_bot/venv/bin/python /opt/tg\_bot/bot.py

Restart=on-failure

RestartSec=5

[Install]

WantedBy=multi-user.target

### ****5. Дополнительные улучшения****

1. **Автоматический перевод пользователей на бесплатный тариф.**
2. **Уведомления в Telegram-боте при превышении лимита трафика.**
3. **Резервное копирование базы данных и RSA ключей.**
4. **Мониторинг работы VPN-сервиса.**
5. **Подготовка к будущему масштабированию.**

Если всё понятно, можно приступать к реализации. 🚀

**Ниже приведён пошаговый туториал по реализации проекта VPN-сервера с Telegram-ботом, а также пример readme-файла для размещения на Git.  
Обратите внимание, что в данном руководстве мы не разбираем сам код bot.py (это запланировано на следующий этап). Здесь мы фокусируемся на настройке SSH, сервера OpenVPN, RSA-ключей, базы данных, окружения для бота и сопутствующих сервисов.**

# ****Пошаговая инструкция по реализации проекта****

## 1. Предварительные требования

1. **Сервер с Ubuntu/Debian**  
   Рекомендуется использовать свежую версию (например, Ubuntu 22.04 LTS или Debian 11).
2. **Доступ по SSH**  
   Убедитесь, что у вас есть root-доступ или права sudo на сервере.
3. **Установленный Python 3**  
   Обычно в современных дистрибутивах Python3 уже есть в комплекте.

## 2. Установка и настройка OpenSSH

1. **Обновите пакеты**:

sudo apt update

1. **Установите и запустите OpenSSH**:

sudo apt install openssh-server -y

sudo systemctl enable ssh

sudo systemctl start ssh

1. **Настройте брандмауэр (ufw) для SSH**:

sudo ufw allow 22

sudo ufw enable

После этого ufw будет блокировать все порты, кроме разрешённых (в данный момент 22/tcp).

## 3. Установка и базовая настройка OpenVPN

1. **Установите OpenVPN и Easy-RSA**:

sudo apt update

sudo apt install openvpn easy-rsa -y

1. **Создайте рабочий каталог easy-rsa**:

make-cadir /etc/openvpn/easy-rsa

cd /etc/openvpn/easy-rsa

1. **Инициализация PKI** (Public Key Infrastructure):

./easyrsa init-pki

1. **Создайте CA (Certificate Authority)**:

./easyrsa build-ca nopass

Файл ca.crt будет создан в папке pki/.

1. **Создайте серверный ключ и сертификат**:

# Генерация запроса на сертификат

./easyrsa gen-req server nopass

# Подпись сертификата

./easyrsa sign-req server server

1. **Скопируйте необходимые файлы в /etc/openvpn/server/**

cd pki

sudo cp ca.crt issued/server.crt private/server.key /etc/openvpn/server/

sudo cp dh.pem /etc/openvpn/server/ # если вы генерировали DH

Файл dh.pem можно получить командой ./easyrsa gen-dh, если вы решили использовать классическую схему Diffie-Hellman.

1. **Создайте конфигурационный файл /etc/openvpn/server/server.conf** (пример из ТЗ):

sudo nano /etc/openvpn/server/server.conf

Пример содержимого (домен/внешний IP замените на свой):

local 45.76.38.87

port 1194

proto udp

dev tun

ca ca.crt

cert server.crt

key server.key

dh dh.pem

auth SHA512

tls-crypt tc.key

topology subnet

server 10.8.0.0 255.255.255.0

server-ipv6 fddd:1194:1194:1194::/64

push "redirect-gateway def1 ipv6 bypass-dhcp"

ifconfig-pool-persist ipp.txt

push "dhcp-option DNS 1.1.1.1"

push "dhcp-option DNS 1.0.0.1"

push "block-outside-dns"

keepalive 10 120

user nobody

group nogroup

persist-key

persist-tun

verb 3

crl-verify crl.pem

explicit-exit-notify

1. **Разрешите порт 1194/udp через ufw**:

sudo ufw allow 1194/udp

1. **Активируйте и запустите OpenVPN**:

sudo systemctl enable openvpn@server

sudo systemctl start openvpn@server

## 4. Настройка RSA для клиентов

При создании нового клиента вам нужно будет:

1. Перейти в папку Easy-RSA:

cd /etc/openvpn/easy-rsa

1. Создать клиентский сертификат:

./easyrsa build-client-full <client\_name> nopass

1. Скопировать готовые файлы (ключ и сертификат) в нужный каталог (например, /etc/openvpn/client/<client\_name>).

## 5. Организация структуры файлов и каталогов

По техническому заданию структура должна выглядеть так:

/etc/openvpn/

├── server/

│ └── server.conf

├── easy-rsa/

│ └── ... (ключи, сертификаты, pki, etc.)

└── client/

└── ... (сгенерированные клиентские .ovpn-файлы)

/opt/tg\_bot/

├── bot.py (код бота, создадим позже)

├── generate\_user\_config.sh (скрипт генерации .ovpn)

├── base/

│ └── vpn\_users.db (файл SQLite-базы данных)

└── logs/ (логи работы бота)

## 6. Настройка Fail2Ban (рекомендуется для безопасности)

1. **Установите fail2ban**:

sudo apt install fail2ban -y

1. **Настройте фильтры и jail.local** (по необходимости).  
   Обычно настройка по умолчанию уже обеспечивает базовую защиту от брутфорса по SSH.

## 7. Настройка базы данных (SQLite)

1. **Установите SQLite** (если не установлено):

sudo apt install sqlite3 -y

1. **Создайте папку для базы** (согласно ТЗ):

sudo mkdir -p /opt/tg\_bot/base

1. **Инициализируйте базу** vpn\_users.db (можно запустить sqlite3 и выполнить команды):

cd /opt/tg\_bot/base

sqlite3 vpn\_users.db

Внутри консоли SQLite выполните:

CREATE TABLE users (

id INTEGER PRIMARY KEY AUTOINCREMENT,

telegram\_id INTEGER UNIQUE,

username TEXT,

tariff TEXT,

traffic\_left INTEGER,

expiry\_date TEXT,

payment\_status TEXT,

config\_path TEXT

);

CREATE TABLE traffic (

id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER,

date TEXT,

bytes\_used INTEGER,

FOREIGN KEY(user\_id) REFERENCES users(id)

);

Выйдите из консоли sqlite> командой .quit.

## 8. Создание окружения для Telegram-бота

1. **Создайте и активируйте виртуальное окружение**:

sudo mkdir -p /opt/tg\_bot

cd /opt/tg\_bot

python3 -m venv venv

source venv/bin/activate

1. **Установите нужные библиотеки**:

pip install aiogram

Примечание: sqlite3 входит в стандартную библиотеку Python, отдельно устанавливать не нужно.

1. **Получите токен бота**  
   Через @BotFather в Telegram создайте бота и получите BOT\_TOKEN. Сохраните его для следующего этапа, где будет писаться код bot.py.

## 9. Скрипт generate\_user\_config.sh

Создайте скрипт для генерации .ovpn-файла (как в ТЗ):

#!/bin/bash

CLIENT\_NAME=$1

OUTPUT\_DIR="/etc/openvpn/client/"

EASY\_RSA\_DIR="/etc/openvpn/easy-rsa"

# Создаём сертификат для нового клиента

cd $EASY\_RSA\_DIR

./easyrsa build-client-full $CLIENT\_NAME nopass

# Предположим, что мы используем стандартный шаблон .ovpn,

# однако в ТЗ сказано копировать crt в файл .ovpn —

# логика может быть расширена при необходимости

cp $EASY\_RSA\_DIR/pki/issued/$CLIENT\_NAME.crt $OUTPUT\_DIR/$CLIENT\_NAME.ovpn

Убедитесь, что скрипт исполнен:

sudo chmod +x /opt/tg\_bot/generate\_user\_config.sh

## 10. Создание сервиса для Telegram-бота (Systemd)

1. **Создайте файл сервиса**:

sudo nano /etc/systemd/system/vpn\_bot.service

1. **Пример содержимого**:

[Unit]

Description=HideWay Bot VPN Service

After=network.target

[Service]

Type=simple

User=root

WorkingDirectory=/opt/tg\_bot

ExecStart=/opt/tg\_bot/venv/bin/python /opt/tg\_bot/bot.py

Restart=on-failure

RestartSec=5

[Install]

WantedBy=multi-user.target

1. **Активируйте и запустите сервис**:

sudo systemctl enable vpn\_bot.service

sudo systemctl start vpn\_bot.service

Статус проверяется командой:

sudo systemctl status vpn\_bot.service

На этом этапе у нас готовы:

* SSH доступ и брандмауэр (ufw).
* OpenVPN-сервер и базовые настройки безопасности (RSA, fail2ban).
* База данных SQLite с необходимой структурой.
* Папка для бота и виртуальное окружение Python.
* Скрипт генерации .ovpn-файлов для клиентов.
* Systemd-сервис, который мы задействуем, когда напишем код бота.

# ****Листинг для README.md****

Ниже — пример файла README.md, который вы можете разместить в корне вашего репозитория (или внутри /opt/tg\_bot, если хотите вести документацию там же). Этот шаблон отражает всю важную информацию о проекте, структуре и шагах установки. При желании добавляйте/меняйте разделы.

# HideWay VPN + Telegram Bot

\*\*HideWay\*\* — это инфраструктура, позволяющая предоставлять бесплатные и в перспективе платные VPN-соединения пользователям через Telegram-бота.

## 1. Описание проекта

- \*\*VPN-сервер\*\* на базе OpenVPN.

- \*\*Управление пользователями\*\* и статистикой трафика через базу данных SQLite.

- \*\*Telegram-бот\*\* для выдачи пользователям готовых `.ovpn`-конфигураций, уведомлений о лимите трафика и получения помощи по настройке.

## 2. Ключевые функции

1. \*\*Бесплатный тариф\*\* (10 ГБ/мес) и возможность добавлять платные тарифы в будущем.

2. \*\*Генерация `.ovpn`\*\* конфигурационных файлов для каждого пользователя.

3. \*\*Автоматическое переключение\*\* на бесплатный тариф после истечения оплаченного лимита.

4. \*\*RSA-ключи\*\* для шифрования (Easy-RSA).

5. \*\*Безопасность\*\*:

- Шифрование AES-256-CBC, аутентификация SHA512.

- Fail2Ban для защиты от брутфорса.

- ufw для ограничения входящих подключений.

6. \*\*Telegram-бот\*\*:

- `/start` — приветственное сообщение.

- `/help` — инструкции по настройке VPN.

- `/getconfig` — получение `.ovpn` файла.

- `/gettraffic` — статистика трафика.

## 3. Структура файлов и каталогов

```plaintext

/etc/openvpn/

├─ server/

│ └─ server.conf # Конфиг OpenVPN

├─ easy-rsa/ # Файлы RSA, pki

└─ client/ # Сгенерированные .ovpn-файлы

/opt/tg\_bot/

├─ bot.py # Код Telegram-бота (написать позже)

├─ generate\_user\_config.sh # Скрипт генерации ovpn

├─ base/

│ └─ vpn\_users.db # SQLite-база данных

└─ logs/ # Логи работы бота

## 4. Установка и настройка

### 4.1 OpenSSH

sudo apt update && sudo apt install openssh-server -y

sudo ufw allow 22

sudo ufw enable

### 4.2 OpenVPN и Easy-RSA

sudo apt update && sudo apt install openvpn easy-rsa -y

make-cadir /etc/openvpn/easy-rsa

cd /etc/openvpn/easy-rsa

./easyrsa init-pki

./easyrsa build-ca nopass

./easyrsa gen-req server nopass

./easyrsa sign-req server server

# ...

(См. полный список команд в [docs/INSTALL.md] или выше.)

### 4.3 База данных

sudo apt install sqlite3 -y

cd /opt/tg\_bot/base

sqlite3 vpn\_users.db

CREATE TABLE users (

id INTEGER PRIMARY KEY AUTOINCREMENT,

telegram\_id INTEGER UNIQUE,

username TEXT,

tariff TEXT,

traffic\_left INTEGER,

expiry\_date TEXT,

payment\_status TEXT,

config\_path TEXT

);

CREATE TABLE traffic (

id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER,

date TEXT,

bytes\_used INTEGER,

FOREIGN KEY(user\_id) REFERENCES users(id)

);

### 4.4 Скрипт генерации конфигураций

#!/bin/bash

CLIENT\_NAME=$1

OUTPUT\_DIR="/etc/openvpn/client/"

EASY\_RSA\_DIR="/etc/openvpn/easy-rsa"

cd $EASY\_RSA\_DIR

./easyrsa build-client-full $CLIENT\_NAME nopass

cp $EASY\_RSA\_DIR/pki/issued/$CLIENT\_NAME.crt $OUTPUT\_DIR/$CLIENT\_NAME.ovpn

### 4.5 Telegram-бот

1. **Создать бота** через BotFather и получить токен.
2. **Разместить код** bot.py в /opt/tg\_bot/.
3. **Создать виртуальное окружение**:

cd /opt/tg\_bot

python3 -m venv venv

source venv/bin/activate

pip install aiogram

1. **Запуск бота**:

python bot.py

Или через systemd:

[Unit]

Description=HideWay Bot VPN Service

After=network.target

[Service]

Type=simple

User=root

WorkingDirectory=/opt/tg\_bot

ExecStart=/opt/tg\_bot/venv/bin/python /opt/tg\_bot/bot.py

Restart=on-failure

RestartSec=5

[Install]

WantedBy=multi-user.target

## 5. Безопасность

* **UFW**: Разрешены только 22/tcp (SSH) и 1194/udp (OpenVPN).
* **Fail2Ban**: Блокирует подозрительные IP-адреса с большим числом неудачных попыток.

## 6. Дополнительно

* Возможность масштабировать проект: добавить **платные тарифы**, интегрировать **платёжные системы**, расширить **функционал бота** (логирование, аналитика и т.д.).
* Планируется автоматическая интеграция с **Cron**/**systemd timers** для проверки остатков трафика.

**Внимание**: Данный проект подразумевает предоставление VPN-услуг. Перед развёртыванием обязательно проверьте юридические аспекты и соответствие локальным законам.  
**Поддержка**: Для получения дополнительной информации пишите администратору или смотрите раздел Issues на GitHub.

Таким образом, вы получили:

1. **Подробное руководство** по установке и настройке всех компонентов: от OpenSSH до системных сервисов и базы данных.
2. **Шаблон** README.md в красивом формате Markdown для удобного размещения на GitHub/GitLab или в локальном репозитории.

На следующем этапе вы сможете дописать/дооформить bot.py, где будет логика Telegram-бота, взаимодействие с базой данных, проверка трафика и генерация .ovpn файлов.