VPNFace Lite

Выпуск 1.0

апр. 18, 2019

Оглавление

VPNFace Lite - набор скриптов для упрощённого разворачивания минимальной инфраструктуры OpenVPN серверов и панель управления клиентскими ключами доступа.

Глава 1

Быстрая установка

OS Ubuntu 18.04/18.10, root

После ssh подключения, используйте скрипт установки через wget

```
`sh wget -qO- https://raw.githubusercontent.com/abrakadobr/vpnface_lite/master/install.sh
| bash `
```

или curl

```
`sh curl -o- https://raw.githubusercontent.com/abrakadobr/vpnface_lite/master/install.sh
| bash `
```

После выдачи приглашения, завершите установку через веб интерфейс.

Глава 2

Online документация:

http://vpnface-lite.readthedocs.io/

Глава 3

Changelog

Check CHANGELOG.md

Содержание:

3.1 Установка

3.1.1 Системные требования

Для простой установки требуется OS Ubuntu/Debian, root доступ, доступ в интернет, однако обязательные зависимости проекта - это `iptables-persistent`, `tor`, `nodejs`, `easy-rsa`, `openvpn`. Таким образом в режиме *ручной установки* систему можно установить на любое устройство-систему, поддерживающие необходимые пакеты.

3.1.2 Простая установка (по умолчанию)

Простая установка расчитана на быстрое разворачивание минимальной впн инфрастуктуры на чистом сервере. Процесс разделён на несколько шагов, для гарантии выполнения каждого предыдущего.

Старт

После ssh подключения, используйте скрипт установки через wget

```
wget -qO- https://raw.githubusercontent.com/abrakadobr/vpnface_lite/v1.1.4/install.sh |
bash
```

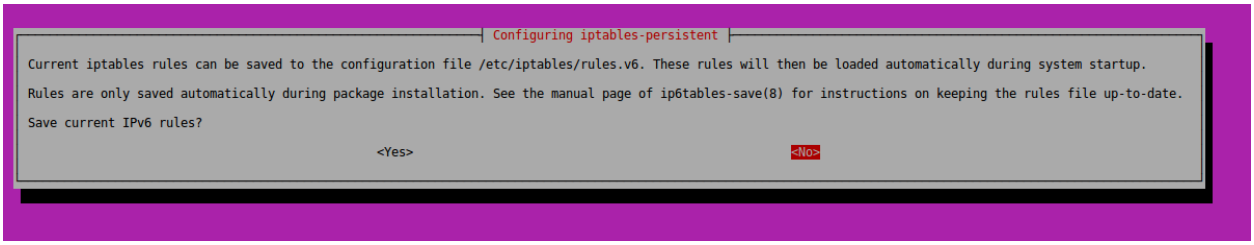
или curl

```
curl -o- https://raw.githubusercontent.com/abrakadobr/vpnface_lite/v1.1.4/install.sh |
bash
```

Скрипт устанавливает пакеты `iptables-persistent`, `easy-rsa`, `openvpn`, `git`, `tor` и `nginx` из системных репозиториев, после чего устанавливает `node version manager` и через него устанавливает `nodejs v10` и пакеты `forever` и `forever-service`. Скрипт клонирует репозиторий проекта, запускает в нём установку зависимостей `npm`, устанавливает сервис *vpnface_lite* и стартует его.

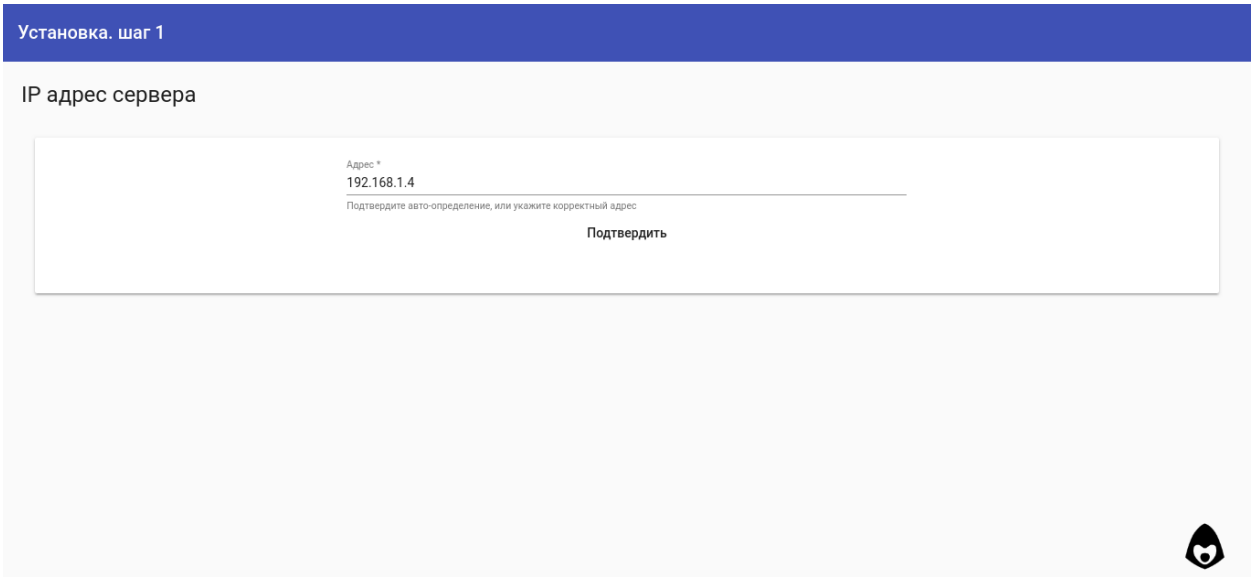
По умолчанию скрипт клонирует файлы проекта в `/opt/vpnface_lite` и хранит данные по пути `/opt/vpnface_ca`.

Во время установки выйдут 2 вопроса от пакета `iptables-persistent` о сохранении текущих настроек. Если у вас чистый сервер без каких-либо предварительных настроек - на оба можно ответить `NO` - в дальнейшем установка автоматически обновит и сохранит конфигурации. Если у вас сервер имеет какие-то настройки `iptables` - заранее позаботьтесь о их сохранности.



По завершению стартовых действия скрипт выводит приглашение завершить установку через веб интерфейс.

Шаг 1

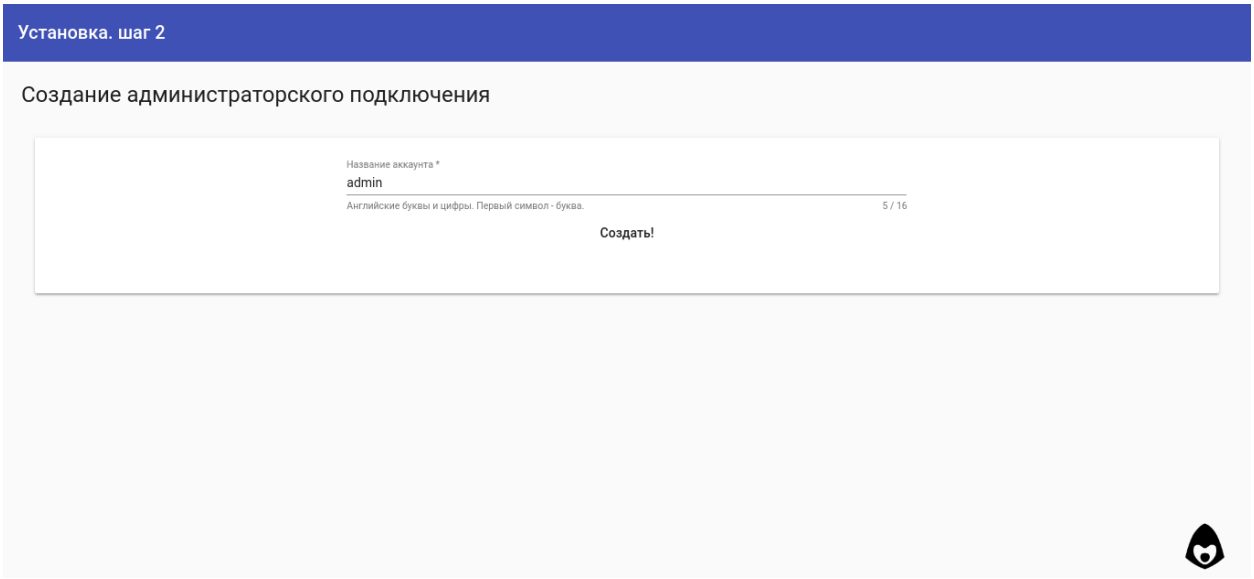


Веб интерфейс запрашивает у пользователя публичный ip-адрес сервера. Скрипт установки пытается определяет адрес для автоматической подстановки через данные http запроса к веб-интерфейсу. После подтверждения ip адреса, скрипт через консоль (`ip -4 -o address`) ищет интерфейс с указаным айпи адресом и сохраняет сетевые данные в json файл `ip.json`.

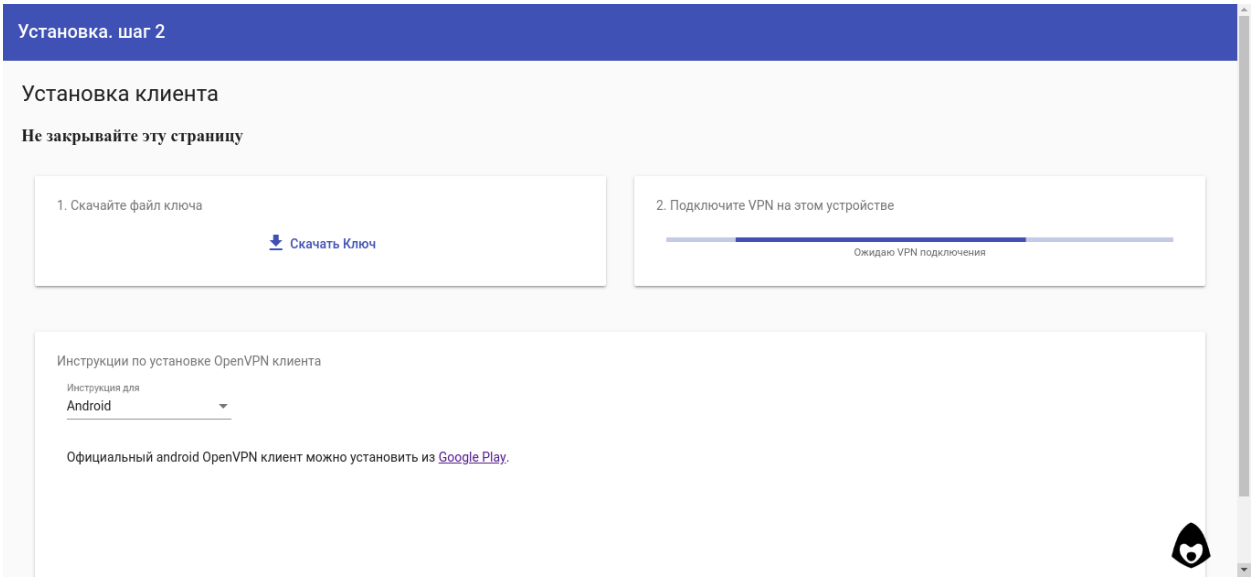
Далее создаётся Административный VPN сервер, и инсталлятор переходит на следующий шаг:

Шаг 2

На втором шаге веб установки пользователю предлагается создать ключ для подключения к Административному VPN серверу.



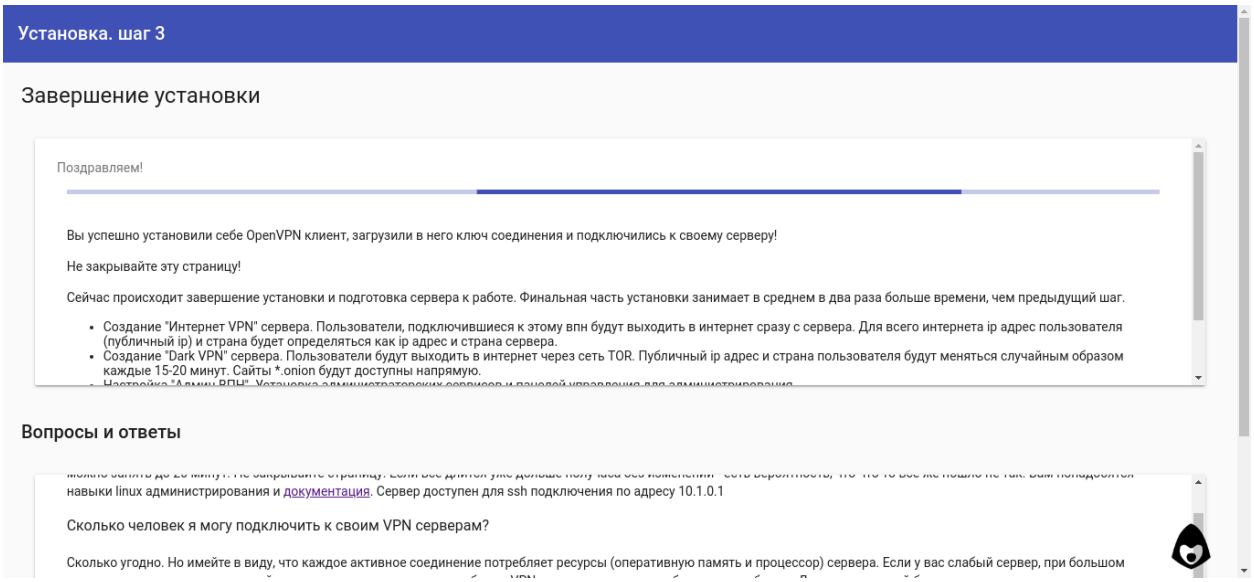
После чего инсталятор переходит в режим ожидания VPN подключения, предлагая пользователю скачать и установить OpenVPN клиент, и созданный на предыдущем шаге ключ соединения.



В это время скрипт зацикленно делает `jsonp` запрос на адрес `http://10.1.0.1/api/ping` ожидая ответа, пока пользователь устанавливает и подключает VPN. Когда ответ успешно получен, скрипт автоматически переходит к следующему шагу по внутреннему vpn адресу `http://10.1.0.1:8808`

Шаг 3

При первом переходе по внутреннему адресу, скрипт автоматически запускает завершение установки.



Устанавливаются Интернет VPN и Dark VPN сервера, конфигурируются `tor` и `nginx`. Настраивается файрвол `iptables`, и так как на этом этапе мы уже уверены, что пользователь успешно подключился через администраторский vpn, на уровне `iptables` сервер закрывается от любого внешнего доступа, кроме трёх впн портов и конфигурация файрвола сохраняется с помощью `iptables-persistent` пакета.

По завершению всех действий пользователь автоматически редиректится в основную панель управления по адресу `http://10.1.0.1`.

Установка с настройкой сертификатов или путей

Один из случаев полу-автоматической установки - установка на чистый сервер, с необходимостью настроить данные сертификации или пути установки. В этом случае достаточно выполнить следующие шаги:

- скачать файлы проекта `git clone https://github.com/abrakadobr/vpnface_lite.git /etc/vpnface_lite`
- отредактировать файл `/etc/vpnface_lite/install.sh` с указанием необходимого пути установки и отключения выкачивания файлов
- отредактировать файл `/etc/vpnface_lite/conf.js` с указанием необхомого пути корневой директории данных, и данных сертификатов (*Файл конфигурации conf.js*)
- после редактирования запустить файл `/etc/vpnface_lite/install.sh` и продолжить простую установку (*Шаг 1*)

3.1.3 Ручная установка

В случае, если конфигурация автоматической установки не устраивает, существует несколько возможных путей конфигурации. Помимо скачивания файлов проекта для полноценной работы системы требуется:

- Установить и настроить пакет `tor` (*Конфигурация TOR*)
- При необходимости установить и настроить пакет `nginx` (*Конфигурация NGINX*)
- Установить `NodeJS v10` (*NodeJS*)

- Если необходимо, создать и настроить службу `vpnface_lite` для файла `vpnface_lite/server.js` (*Сервис vpnface_lite*)
- Настроить файрвол `iptables/ufw/other` на правильную маршрутизацию трафика и доступы (*Iptables*)

Установка панели управления на работающий сервер

Другой случай, в котором автоматическая настройка системы может не подойти - это подключение панели управления ключами к существующим и уже работающим OpenVPN серверам, или установка системы на уже работающий в определённой конфигурации сервер. В этом случае шаги будут следующие:

- скачать файлы проекта `git clone https://github.com/abrakadobr/vpnface_lite.git` в необходимую директорию
- отредактировать файл `vpnface_lite/conf.js` с указанием необхомого пути корневой директории данных, данные сертификатов и серверов в этом случае не имеют накакого значения. (*Файл конфигурации conf.js*)
- настроить директорию данных по стуктуре VPNFace Lite. (*Директория данных*)
- дополнить easy-rsa скрипты для корректной блокировки-разблокировки ключей (*Центр ключей <S>_ca*)
- при необходимости, перенастроить OpenVPN сервера, с указанием порта менеджмента (*Серверная структура /etc/openvpn/*)
- создать файл `ip.json` в папке данных с описанием сетевой стуктуры. (*Сетевые настройки ip.json*)
- создать файл `servers.json` в папке данных с описанием серверов (*База данных servers.json*)
- настроить проксирование веб-панели управления с порта 8808 на необходимые. (*Конфигурация NGINX*)
- установить `nodejs 10+` и настроить запуск сервера по необходимости в виде службы, или как-то ещё. (*Сервис vpnface_lite*)

VPNFace при старте проверяет установку по файлам `ip.json` и `servers.json`, и в зависимости от этого переходит в режим установки или обычной работы. Таким образом, создание этих файлов с корректными данными позволит пропустить автоматическую установку, и запустить систему в рабочем режиме. Подробную информацию о *.json файлах, структуре директорий и дополнительных скриптах смотрите в разделе *Технические детали*

3.2 Использование

3.2.1 VPN сервера

VPN соединение - шифрованное соединение напрямую между клиентом и сервером. Такие *сетевые туннели* позволяют связать пользователей из разных геолокаций планеты и интернет операторов в общую защинённую от доступа «из-вне» сеть.

По умолчанию, VPNFace Lite создаёт три OpenVPN сервера для различных задач:

Админ VPN

VPN предназначен для обеспечения безопасности серверного администрирования, поэтому он не подключён к общей интернет сети, и при подключении с установками клиента по умолчанию - у клиента отсутсвует доступ в интернет.

В этой vpn подсети (10.1.0.0/24) сервер полностью доступен по адресу 10.1.0.1

`ssh 10.1.0.1` для ssh соединения, `http://10.1.0.1` - для доступа в панель управления ключами, `http://10.1.0.1:81` - докментация.

Интернет VPN

VPN предназначен для доступа в общий интернет через сервер. Это позволяет держать зашифрованными все данные между клиентом и сервером, и подменяет «интернет публичные» данные клиента данными сервера, так как расшифрованый интернет трафик в *публичный интернет* выходит уже с сервера.

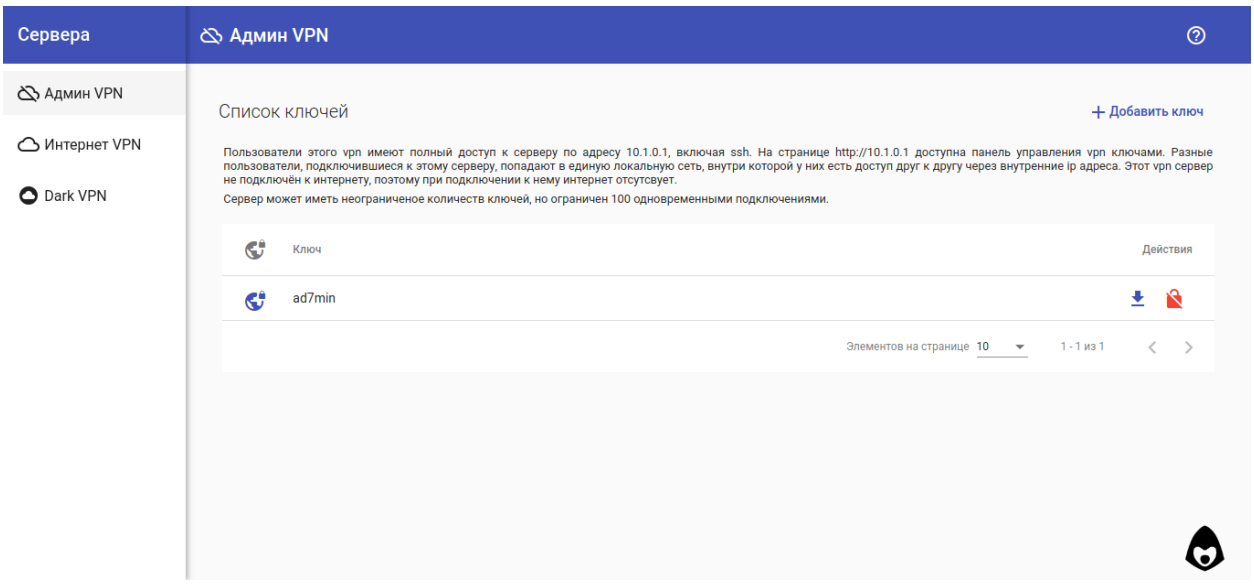
DNS запросы пользователя отправляются на публичные Google DNS 8.8.8.8 и 8.8.4.4

Dark (TOR) VPN

Dark VPN *(англ. тёмный, чёрный)*, использует TOR даркнет для доступа в интернет. Пользовательский трафик проходит шифровано до сервера через vpn туннель, откуда перенаправляется в TOR сеть. Перенаправляются так же и DNS запросы, поэтому любые *.onion сайты работают автоматически, прозрачно для пользователя.

Настройка TOR по умолчанию описана в *Конфигурация TOR*, ряд стран по умолчанию поставлен в блокированные в качестве выходных узлов.

3.2.2 Панель управления ключами



Панель упрощена до минимума, и позволяет создавать, блокировать и разблокировать клиентские ключи, а так же скачивать готовую конфигурацию в виде ovpn файла.

Панель написана на `AngularJS 6`, представляет из себя веб приложение для использования vpnface_lite api endpoints и идёт по умолчанию в комплекте с серверными файлами в скомпилированном виде.

Исходные коды так же доступны в отдельном репозитории https://github.com/abrakadobr/vpnface_lite_ng.git

3.2.3 JSON API

VPNFace Lite предоставляет набор json api endpoints котрые могут быть использованы отдельно, без панели управления через http запросы.

Endpoints, использующиеся при установке:

GET	/api/logs	получить последние логи
GET	/api/status	текущий статус.
GET	/api/finilize	запустить завершение установки
POST	/api/confirmip	установить ip и стартовать установку POST:ip
JSONP	/api/ping	во время установки подтверждает подключение vpn

Endpoints, использующиеся при работе:

GET	/api/logs	получить последние логи
GET	/api/status	текущий статус.
GET	/api/ovpn/:key	скачать ovpn ключ
POST	/api/gencli	создать новый ключ POST:srv, POST:cli key=cli@srv
GET	/api/lock/:key	заблокировать ключ
GET	/api/unlock/:key	разблокировать ключ
GET	/api/vpnlist	полный список серверов (файл servers.json)
GET	/api/server/:code	полная информация о отдельном сервере

параметр `:code` - код сервера <S> (*Директория данных*)

параметр `:key` - составная строка вида `client@server`. к примеру для клиентского ключа `admin` с сервера `adm` параметр `:key` будет равен `admin@adm`

3.3 Технические детали

3.3.1 Общая информация

VPNFace Lite не использует никаких специальных баз данных, не занимается синхронизацией с центрами сертификтов, и хранит все данные в двух json файлах.

Во время *Простая установка (по умолчанию)*, доставляются и конфигурируются необходимые пакеты из официальных репозиториев, после установки VPNFace Lite отвечает только за управление ключами vpn серверов.

3.3.2 VPN подсети

По умолчанию VPNFace Lite создаёт три OpenVPN сервера со следующими сетвыми настройками:

VPN	Подсеть	Интерфейс	Адрес сервера
Админ VPN	10.1.0.0/24	tun0	10.1.0.1
Dark VPN	10.2.0.0/24	tun1	10.2.0.1
Inet VPN	10.3.0.0/24	tun2	10.3.0.1

На уровне `iptables` при *Простая установка (по умолчанию)* сети настраиваются следующим образом:

Админ VPN: Полный доступ к серверу по адресу 10.1.0.1, отсутсвие интернета.

Dark VPN: DNS запросы перенаправляются на TorDNS на порту 5300, весь tcp трафик перенаправляется в тор на порту 9040, остальной udp трафик никуда не уходит.

Inet VPN: В качестве DNS серверов используются Google DNS 8.8.8.8, и 8.8.4.4. Весь сетевой трафик через NAT отправляется в интернет.

3.3.3 Конфигурация TOR

По умолчанию TOR настраивается по следующей конфигурации:

`vpnface_lite/tpl/tor.conf`

```
DataDirectory /var/lib/tor

VirtualAddrNetworkIPv4 12.0.0.0/16
AutomapHostsOnResolve 1
TransPort 10.2.0.1:9040
DNSPort 10.2.0.1:5300

StrictNodes 1
ExcludeExitNodes
«{AF},{AL},{DZ},{AI},{BH},{BY},{BT},{BW},{BR},{KH},{CF},{TD},{CL},{CN},{MO},{CG},{CD},{DJ},{EG}
```

3.3.4 Конфигурация NGINX

По умолчанию NGINX настраивается по следующей конфигурации:

`vpnface_lite/tpl/nginx.conf`

```
#панель управления
server {
        listen 10.1.0.1:80;
    access_log /var/log/nginx/vpnface_lite.proxy.log;
    location / {
        proxy_pass http://127.0.0.1:8808;
    }
}
#документация
server {
        listen 10.1.0.1:81;
    access_log /var/log/nginx/vpnface_lite.docs.log;
    root /opt/vpnface_lite/cdocs;
```

```
        index index.html;

        location / {
            default_type «text/html»;
            try_files $uri.html $uri $uri/ /index.html;
        }
    }
```

3.3.5 Директория данных

По умолчанию VPNFace Lite хранит данные в папке `/opt/vpnface_ca`. Для каждого сервера с кодом <S> создаются две директории: `<S>_ca` и `<S>_cli`. Так же в этой директории хранятся файлы `ip.json` и `servers.js`.

Значения <S> при *Простая установка (по умолчанию)*

VPN	<S>
Админ VPN	adm
Dark VPN	darknet
Inet VPN	inet

Центр ключей <S>_ca

Директория создаётся коммандой `make-cadir DIR` из пакета `easy-rsa` и дополняется следующими скриптами:

`build-crl` из шаблона `vpnface_lite/tpl/build-crl.sh` для генерации файла блокировок.

```
    #!/bin/sh

    # revoke a certificate, regenerate CRL,
    # and verify revocation

    if [ -z «$KEY_DIR» ]; then
        echo „Please source the vars script first (i.e. «source ./vars»)“
        exit 1
    fi

    CRL=»crl.pem»

    cd «$KEY_DIR»
    # set defaults
    export KEY_CN=»«
    export KEY_OU=»«
    export KEY_NAME=»«
    # required due to hack in openssl.cnf that supports Subject Alternative Names
    export KEY_ALTNAMES=»«
    # generate a new CRL – try to be compatible with
    # intermediate PKIs
```

$OPENSSL ca -gencrl -out «$CRL» -config «$KEY_CONFIG»

`revoke-key` из шаблона `vpnface_lite/tpl/revoke-key.sh` с заменой ключа #MPORT на менеджмент порт openvpn сервера для блокировки пользователя.

```
#!/bin/bash

if [ $# -ne 1 ]; then
        echo «usage: voke-key <cert-name-base>»;
        exit 1
fi
if [ -z «$KEY_DIR» ]; then
        echo „Please source the vars script first (i.e. «source ./vars»)“
        exit 1
fi

KEYS_INDEX=$KEY_DIR/index.txt
LINE=‘grep «/CN=$1/» $KEYS_INDEX‘
COLS_NUM=‘echo $LINE | awk -F“ „ „{print NF;}“‘

echo $COLS_NUM

if [[ $COLS_NUM -eq 5 ]] && [[ $LINE == V* ]]; then

        ./revoke-full $1
        {
                sleep 3
                echo kill $1
                sleep 3
                echo exit
        } | telnet localhost #MPORT
        echo «Certificate revoked successfully.»
        exit 0;
elif [[ $COLS_NUM -eq 6 ]] && [[ $LINE == R* ]]; then
        echo «Client certificate is already revoked.»
        exit 0;
else
        echo «Error; Key index file may be corrupted.»
        exit 1;
fi
```

и `voke-key` с шаблона `vpnface_lite/tpl/voke-key.sh` для разблокирования пользовательского ключа

```
#!/bin/bash

if [ $# -ne 1 ]; then
        echo «usage: voke-key <cert-name-base>»;
        exit 1
```

```
fi
if [ -z «$KEY_DIR» ]; then
        echo „Please source the vars script first (i.e. «source ./vars»)“
        exit 1
fi

KEYS_INDEX=$KEY_DIR/index.txt
NLINE=‘grep -n «/CN=$1/» $KEYS_INDEX‘
LINE=‘grep «/CN=$1/» $KEYS_INDEX‘
LINE_NUM=‘echo $NLINE | cut -f1 -d:‘
COLS_NUM=‘echo $LINE | awk -F“ „ „{print NF;}“‘

echo $COLS_NUM

if [[ $COLS_NUM -eq 6 ]] && [[ $LINE == R* ]]; then
        COL2=‘echo $NLINE | awk „{print $2}“‘
        COL4=‘echo $NLINE | awk „{print $4}“‘
        COL5=‘echo $NLINE | awk „{print $5}“‘
        COL6=‘echo $NLINE | awk „{print $6}“‘
        echo -e «Vt$COL2tt$COL4t$COL5t$COL6» >> $KEYS_INDEX
        sed -i «${LINE_NUM}d» $KEYS_INDEX
        ./build-crl
        echo «Certificate unrevoked successfully.»
        exit 0;
elif [[ $COLS_NUM -eq 5 ]] && [[ $LINE == V* ]]; then
        echo «Certificate is already unrevoked and active»
        exit 0;
else
        echo «Error; Key index file may be corrupted.»
        exit 1;
fi
```

так же, при автоматическом создании серверов в файле `<S>_ca/vars` производятся настройки данных сертификата.

все ключи и файлы криптографии сохраняются в директории `<S>_ca/keys`.

OVPN генератор <S>_cli

Директория содержит базовый клиентский конфиг сервера, и скрипт для генерации ovpn файлов. Так же тут нахдится директория `files` в которую помещаются сгенерированные клиентские ovpn файлы.

`cli.conf` - шаблон клиентского конфига, при *Простая установка (по умолчанию)* генерируется при создании серверов, с заменой ключей необходимыми данными из шаблона `vpnface_lite/tpl/cli.conf`

```
client
dev tun
proto #PROTO
remote #REMOTE #PORT
```

#GW
nobind
user nobody
group nogroup
persist-key
persist-tun
remote-cert-tls server
key-direction 1
cipher AES-256-CBC
auth SHA256
verb 3

таблица ключей

ключ	значение	Admin VPN	Dark VPN	Inet VPN
#PROTO	протокол tcp/udp	udp	udp	udp
#REMOTE	интернет ip сервера	server_ip	server_ip	server_ip
#PORT	порт openvpn сервера	1194	1195	1196
#GW	параметры DNS/route	—	TOR DNS	GoogleDNS

`cli.sh` - баш скрипт генерации ovpn файла, из шаблона `vpnface_lite/tpl/cli.sh`

#!/bin/bash

#KEY_DIR
#OUTPUT_DIR
#BASE_CONFIG

cat ${BASE_CONFIG}
<(echo -e „<ca>“)
${KEY_DIR}/ca.crt
<(echo -e „</ca>n<cert>“)
${KEY_DIR}/${1}.crt
<(echo -e „</cert>n<key>“)
${KEY_DIR}/${1}.key
<(echo -e „</key>n<tls-auth>“)
${KEY_DIR}/ta.key
<(echo -e „</tls-auth>“)
> ${OUTPUT_DIR}/${1}.ovpn

таблица ключей

ключ	значение
#KEY_DIR	директория <S>_ca/keys - источник ключей
#OUTPUT_DIR	дирекория <S>_cli/files - выходные ovpn
#BASE_CONFIG	файл <S>_cli/cli.conf - шаблон клиентского конфига

Сетевые настройки ip.json

{

```
    «ip»: «интернет ip адрес сервера»,
    «dev»: «интернет сетевой интерфейс, например eth0»
}
```

База данных servers.json

Файл содержит базу данных серверов и клиентов, на которую опирается VPNFace Lite. Если вы производите ручную установку для уже имеющихся серверов (*Ручная установка*), вам требуется при установке сформировать этот файл вручную по следующему формату:

В качестве примера используется сервер S1 с подсетью 10.1.0.1/24, и содержит полный пример конфига, а сервер S2 настроен в минимальном для работы режиме, для примера установки на уже имеющийся сервер, с подготовленной вручную структурой файлов и директорий.

Поля, помеченые * используются только во время создания сервера

```
{
    «S1»: {
        «code»: «S1»,
        «name»: «Имя сервера для панели управления»,
        «desc»: «Описание для панели управления»,
        «logs»: true/false, //* ведёт ли сервер логи
        «friends»: true/false, //* видят ли клиенты друг-друга
        «maxclients»: 100, //* максимальное количество соединений
        «type»: «root», //тип сервера
        «network»: {
            «host»: «0.0.0.0», //* на каком хосте запускать openvpn сервер
            «remote»: «интернет ip сервера», //*
            «intranet»: «10.1.0.0/24», //* vpn подсеть
            «port»: 1194, //* на каком порту vpn сервер ожидает клиентов
            «mport»: 2294, //* порт управления vpn сервером
            «proto»: «udp», //* протокол соединения
            «dev»: «tun0», //сетевой интерфейс vpn сервера на сервере
        },
        «cert»: { //* параметры сертификата для файла S1_ca/vars
            «country»: «US»,
            «province»: «CA»,
            «city»: «City»,
            «org»: «Organisation»,
            «email»: «email@domain.zone»,
            «ou»: «Organization Unit»
        },
        «intranet»: { //* сетевые данные vpn подсети, генерируются при создании
            «networkAddress»: «10.1.0.0»,
            «firstAddress»: «10.1.0.1»,
            «lastAddress»: «10.1.0.254»,
            «broadcastAddress»: «10.1.0.255»,
            «subnetMask»: «255.255.255.0»,
            «subnetMaskLength»: 24,
```

```
                «numHosts»: 254,
                «length»: 256
            },
            «clients»: [ // массив клиентов.
                {
                    «code»: «client1_code», // код ключа
                    «blocked»: true/false, // состояние блокировки
                    «server»: «S1» // код сервера, для удобства
                },{
                    «code»: «client2_code», // код ключа
                    «blocked»: true/false, // состояние блокировки
                    «server»: «S1» // код сервера, для удобства
                }
            ]
        },
        «S2»: {
            «code»: «S2»,
            «name»: «Имя сервера для панели управления»,
            «desc»: «Описание для панели управления»,
            «maxclients»: 100, //* максимальное количество соединений, отображается в
            панели
            «type»: «public», //тип сервера
            «clients»: [ // массив клиентов.
                {
                    «code»: «client1_code», // код ключа
                    «blocked»: true/false, // состояние блокировки
                    «server»: «S1» // код сервера, для удобства
                },{
                    «code»: «client2_code», // код ключа
                    «blocked»: true/false, // состояние блокировки
                    «server»: «S1» // код сервера, для удобства
                }
            ]
        },
        «<S3>»: { ... }
        «<S4>»: { ... }
}
```

3.3.6 Серверная структура /etc/openvpn/

В директории `/etc/openvpn/` хранятся конфигурации и ключи рабочих серверов. Для каждого сервера VPNFace Lite создаёт директорию `<S>` в которуй хранятся файлы ключей, криптографии и блокировок, и файл `<S>.conf` серверной конфигурации, который создаётся по шаблону `vpnface_lite/tpl/server.conf`

```
#LOCAL
port #PORT
proto #PROTO
```

dev #DEV
ca #CA_CRT
cert #SERVER_CRT
key #SERVER_KEY
dh #DH_PEM
#INTRANET
ifconfig-pool-persist /var/log/openvpn/ipp-#SERVER.txt
#GW
#FRIENDS
keepalive 10 120
tls-auth #TA_KEY 0
key-direction 0
cipher AES-256-CBC
auth SHA256
crl-verify #SERVER_CRL
management 127.0.0.1 #MANAGEMENT_PORT
max-clients #MAX_CLIENTS
user nobody
group nogroup
persist-key
persist-tun
#LOG
verb 3
explicit-exit-notify 1

таблица ключей

ключ	значение
#LOCAL	значение <conf.network.host> если не 0.0.0.0
#PORT	порт клиентских соединений
#DEV	сетевой интерфейс
#CA_CRT	<S>/ca.crt
#SERVER_CRT	<S>/<S>.crt
#DH_PEM	<S>/dh2048.pem
#INTRANET	„server <conf.intranet.*>“
#GW	push route/DNS в зависимости от <conf.type>
#FRIENDS	„client-to-client“ если <conf.firends == true>
#SERVER_CRL	<S>/crl.pem
#MANAGEMENT_PORT	„127.0.0.1 <conf.network.mport>“
#MAX_CLIENTS	<conf.maxclients>
#LOG	log-status и log-append в зависимости от <conf.logs>

3.3.7 CONF.TYPE настройка

Параметр `type` в конфигурации оказывает влияние на опции создания OpenVPN серверов и иконку в панели администрирования.

В будущем это будет оказывать влияние на настройки `iptables` но в версии Lite файрвол конфигурируется по предустановленым значениям, так что на iptables влияния нет.

Таблица возможных значений

значение	iptables	server.conf	cli.conf	иконка в панели
public	в интернет	GoogleDNS	–	прозрачное облако
root	на сервер	–	no-push	перечёркнутое облако
dark	в тор	TOR DNS	–	облако в закрашеном круге

3.3.8 Файл конфигурации conf.js

Файл располагается по пути `vpnface_lite/conf.js` и содержит настройки необходимые для постоянной работы, и установки в формате javascript.

В случае запуска VPNFace Lite на работающих серверах без встроенной установки (*Установка панели управления на работающий сервер*), файл должен содержать два параметра: порт апи, и путь к директории данных.

По умолчанию файл так же содержит конфигурации сертификата и серверов для *Простая установка (по умолчанию)*. В режиме *Установка с настройкой сертификатов или путей* этот файл можно отконфигурировать под требуемые значения.

Структура файла:

```
module.exports = {
    dir: „/opt/vpnface_ca“,
    port: 8808,
    cert: CERT_CONFIG<Общий сертификат «по умолчанию»>,
    servers: {
        adm: SERVER_CONFIG<Админ VPN>,
        inet: SERVER_CONFIG<Интернет VPN>,
        dark: SERVER_CONFIG<TOR VPN>,
    }
}
```

Полный пример можно посмотреть в файле по умолчанию

3.3.9 Сервис vpnface_lite

VPNFace Lite по умолчанию устанавливается в виде системного сервиса с названием `vpnface_lite` пакетом `forever_service`. Сервис ведёт логи в `/var/log/vpnface_lite.log`

3.3.10 Iptables

iptables - linux файрвол. При автоматической установке выполняется следующий набор комманд, для установки правил:

На старте установки:

```
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -t nat -F
iptables -t mangle -F
```

iptables -t nat -X
iptables -t mangle -X
iptables -F
iptables -X
echo «net.ipv4.ip_forward=1» > /etc/sysctl.conf
echo «net.ipv6.conf.all.disable_ipv6 = 1» >> /etc/sysctl.conf
echo «net.ipv6.conf.default.disable_ipv6 = 1» >> /etc/sysctl.conf
echo «net.ipv6.conf.lo.disable_ipv6 = 1» >> /etc/sysctl.conf
sysctl -p

На завершении:

iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -m conntrack –ctstate ESTABLISHED,RELATED -j ACCEPT

iptables -A INPUT -p udp –dport 1194 -j ACCEPT
iptables -A INPUT -p udp –dport 1195 -j ACCEPT
iptables -A INPUT -p udp –dport 1196 -j ACCEPT

iptables -A INPUT -s 10.1.0.0/24 -j ACCEPT
iptables -A INPUT -s 10.2.0.0/24 -d 10.2.0.1 -j ACCEPT
iptables -A INPUT -s 10.3.0.0/24 -d 10.3.0.1 -j ACCEPT

iptables -t nat -A POSTROUTING -s 10.3.0.0/24 -o <IP.JSON:DEV> -j MASQUERADE

iptables -t nat -A PREROUTING -i tun1 -p udp –dport 53 -j REDIRECT –to-ports 5300
iptables -t nat -A PREROUTING -i tun1 -p tcp –syn -j REDIRECT –to-ports 9040

iptables -A INPUT -j DROP

Обратите внимание, на опции, устанавливаемые в `/etc/sysctl.conf`

net.ipv4.ip_forward = 1
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1

для возможности роутинга трафика и отключения ip v6.

3.3.11 NodeJS

Для установки nodejs v10 используется `node version manager` https://github.com/creationix/nvm

После установки скприт `install.sh` создаёт линк на 10 версию ноды в общесистемный путь

```
ln -s `whitch node` /usr/sbin
```

3.4 CHANGELOG

Глава 4

Version code

<major>/<minor>/<patch>

major increases on core updates minor - on features patch - on every release, reseting on major increase

Глава 5

current master

- fast fix for OpenVPN 2.4 and EasyRSA 3.0.4 for ubuntu 18.04 and 18.10

Глава 6

v1.1.4

- full installation from conf.js with correct network configuration use npm<ip> packet for networks calculation iptables now configured using this calculation, and <dev> option from conf

  полноценная сетевая настройка и конфигурация из conf.js используется пакет npm<ip> для калькуляции сетевых адресов из указанного в конфигрурации iptables теперь конфигурируются так же с учетом этих параметров, и параметра dev

- installation script now use release branch, as it should be, really

  скрип установки теперь использует release ветку, как, в принципе, и должно быть