



**Функция OpenVPN**

**VP-5x**

# Содержание

1.	Обзор	3
2.	OpenVpn	4
2.1	Установка OpenVPN (ОС Windows )	4
2.2	Создание сертификатов	6
2.2.1	Первоначальная конфигурация	6
2.2.2	Создание сертификата СА	7
2.2.3	Создание файла dh1024.pem	8
2.2.4	Создание сертификата сервера	8
2.2.5	Создание сертификата клиента	8
2.3	Файлы конфигурации	8
2.3.1	Конфигурация сервера	8
2.3.2	Подключение к серверу OpenVPN	17
2.3.3	Конфигурация клиента	19
2.3.4	Создание файла client.tar	23
3.	Настройка телефона	24
3.1	Конфигурация через web	24

## 1. Обзор

**VPN (виртуальная частная сеть)** - это технология, обеспечивающая защищённую (закрытую от внешнего доступа) связь логической сети поверх частной или публичной при наличии высокоскоростного интернета.

Соединение использует виртуальный туннель для реализации шифрования данных, проверки и пользователя аутентификация, которая гарантирует, что данные не были фальсифицированы, реплицированы, проверены.

Что касается выделенной линии, VPN работает в Интернете без чрезмерных затрат. Можно безопасно и экономично передавать частные данные через Интернет. VPN-система включает VPN-сервер, VPN-клиент и виртуальный туннель. На рис.1 показана простая схема VPN.

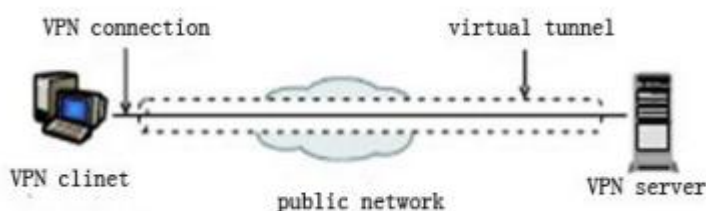


Рис.1

VPN позволяет ip телефону из общедоступной сети иметь безопасный удалённый доступ к частной сети (рис.2)

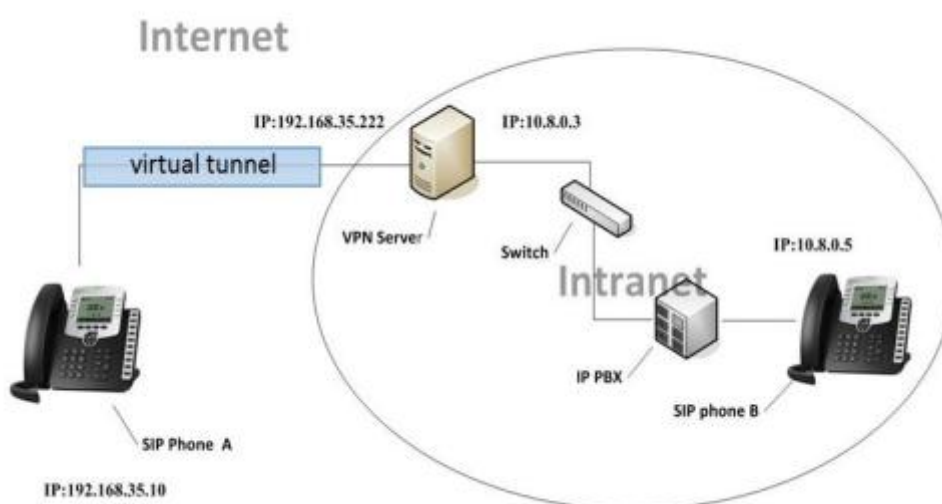


рис.2

## 2. OpenVpn.

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

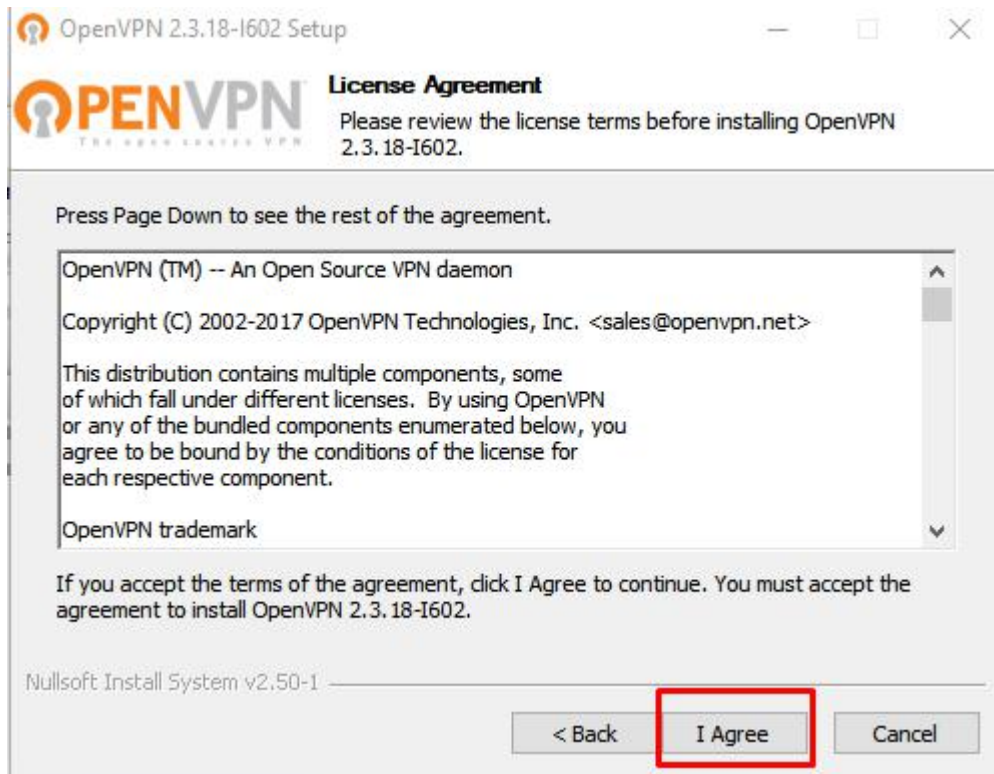
### 2.1 Установка OpenVPN (ОС Windows )

Загрузите и установите **OpenVPN** (например, OpenVPN 2.3.18) и выполните следующие шаги:

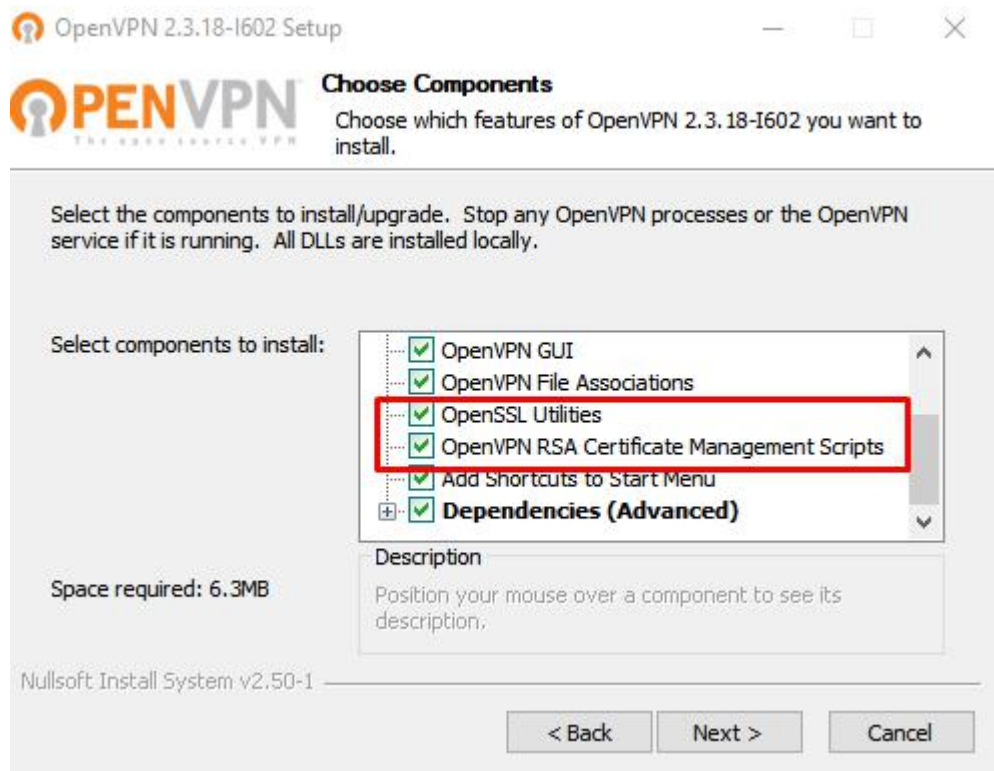
В данном примере, **OpenVPN** устанавливался **C:\Program Files\OpenVPN**.

- Запускаем скачанный файл - нажимаем «**Next**» - «**I Agree**»





- выставляем маркеры «**OpenVPN RSA Certificate Management Scripts**» (нужен для возможности сгенерировать сертификаты) и **OpenSSL Utilities**



«Next» и «Install» — начнётся установка.

В процессе мастер установки может выдать запрос на подтверждение установки виртуального сетевого адаптера — соглашаемся (Install).

После завершения нажимаем «Next» - снимаем галочку «show Readme» - «Finish»

Установка завершена.

## 2.2 Создание сертификатов.

### 2.2.1 Первоначальная конфигурация.

- В папке установки **OpenVPN** (по умолчанию, **C:\Program Files\OpenVPN**) и создаём каталог **SSL**.
- В папке **C:\Program Files\OpenVPN\easy-rsa**, создаём файл **vars.bat**, открываем его на редактирование и приводим к следующему виду:

```
set "PATH=%PATH%;%ProgramFiles%\OpenVPN\bin"
set HOME=%ProgramFiles%\OpenVPN\easy-rsa
set OPENSSL_CONF=C:\Program Files\OpenVPN\easy-rsa\openssl-1.0.0.cnf
set KEY_CONFIG=openssl-1.0.0.cnf
set KEY_DIR=keys
set DH_KEY_SIZE=1024
set KEY_SIZE=1024
set KEY_COUNTRY=RU
set KEY_PROVINCE=Ekaterinburg
set KEY_CITY=Ekaterinburg
set KEY_ORG=Organization
set KEY_EMAIL=i.ivanov@nag.ru
set KEY_CN=NAG
set KEY_OU=NAG
set KEY_NAME=server.domain.ru
set PKCS11_MODULE_PATH=NAG
set PKCS11_PIN=12345
```

\* в каталоге **easy-rsa** уже есть файл **vars.bat.sample** — можно переименовать и использовать его.

\*\* значение **HOME** не меняем, если оставили путь установки программы по умолчанию;

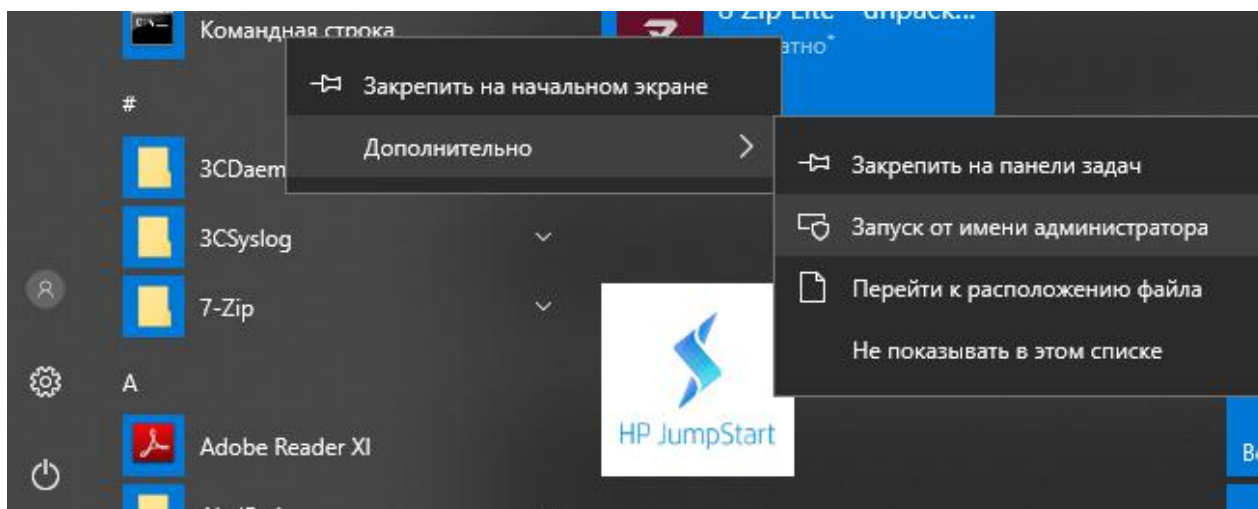
**KEY\_DIR** — каталог, куда будут генерироваться сертификаты;

**KEY\_CONFIG** может быть разным — его лучше посмотреть в файле **vars.bat.sample** или по названию соответствующего файла в папке *easy-rsa*;

**KEY\_NAME** желательно, чтобы соответствовал полному имени VPN-сервера; остальные опции можно заполнить произвольно.

## 2.2.2 Создание сертификата CA .

- Запускаем командную строку (cmd) от имени администратора:



- Переходим в каталог **easy-rsa**:

```
cd C:\\Program Files\\OpenVPN\\easy-rsa
```

- Запускаем файл **vars.bat**

```
vars.bat
```

- Чистим каталоги от устаревшей информации:

```
clean-all.bat
```

- Снова запускаем **vars.bat** (после clean переопределяются некоторые переменные):

```
vars.bat
```

- Генерируем сертификат и на все запросы нажимаем **Enter**.

```
build-ca.bat
```

### 2.2.3 Создание файла dh1024.pem

В командной строке (cmd) введите

```
build-dh
```

### 2.2.4 Создание сертификата сервера

В командной строке необходимо ввести:

```
build-key-server.bat cert
```

*\* где **cert** — имя сертификата; на все запросы нажимаем **Enter**. В конце подтверждаем два раза корректность информации вводом **y**.*

Содержимое	папки		C:\Program
Files\OpenVPN\easy-rsa\keys	переносим	в	C:\Program
Files\OpenVPN\ssl.			

### 2.2.5 Создание сертификата клиента.

Генерируем сертификат пользователя, в командной строке (cmd) вводим:

```
build-key.bat clients
```

*\* на все запросы нажимаем **Enter**, в конце — **y**.*

*\*\* Мы можем на каждого клиента сгенерировать свой сертификат, а можем использовать один на всех. Первый вариант безопаснее, второй — удобнее. Каким пользоваться — решать исходя из личного опыта, требований политики безопасности компании и уровня доверия к пользователям.*

## 2.3 Файлы конфигурации .

### 2.3.1 Конфигурация сервера.

Переносим из папки C:\Program Files\OpenVPN\sample-config в C:\Program Files\OpenVPN\config файл **sample.ovpn**. Переименовываем его в **server.ovpn**. Файл **server.ovpn** открываем на редактирование и приводим к следующему виду:



```

port 12345
proto udp
dev tun
dev-node "VPN Server"
dh "C:\Program Files\OpenVPN\ssl\dh1024.pem"
ca "C:\Program Files\OpenVPN\ssl\ca.crt"
cert "C:\Program Files\OpenVPN\ssl\cert.crt"
key "C:\Program Files\OpenVPN\ssl\cert.key"
server 10.10.0.0 255.255.0.0
max-clients 32
keepalive 10 120
client-to-client
comp-lzo
persist-key
persist-tun
cipher DES-CBC
status "C:\Program Files\OpenVPN\log\status.log"
log "C:\Program Files\OpenVPN\log\openvpn.log"
verb 4
mute 20

```

\* где **port** — сетевой порт (12345 позволит избежать проблем при использовании Интернета в общественных местах, но может быть любым из свободных, например 1723, занятые порты в Windows можно посмотреть командой `netstat -a`);

**dev-node** — название сетевого интерфейса;

**server** — подсеть, в которой будут работать как сам сервер, так и подключённые к нему клиенты.

\*\* так как в некоторых путях есть пробелы, параметр заносится в кавычках.

```

#####
# Which local IP address should OpenVPN
# listen on? (optional)
;local a.b.c.d

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port

```

```

# number for each one. You will need to
# open up this port on your firewall.
# the default port 1194
port 1194

# TCP or UDP server?
#Uncomment the line to enable TCP or UDP
;proto tcp
proto udp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
#Typically, dev tap is used if VPN server is
# running on windows
#tap is for Windows and tun is for Linux
dev tap
;dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one. On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key). Each client

```

```

# and the server must have their own cert and
# key file. The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys. Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in main page).
#Please be sure the filename
#ROOT CA is generated by build-ca, and it is used to verify the
legality of customer certification.
ca "C:\\Program Files\\OpenVPN\\config\\ca.crt"

#The certificate file of server
cert "C:\\Program Files\\OpenVPN\\config\\Cdtmsserver.crt"

#The key of certificate file.
key "C:\\Program Files\\OpenVPN\\config\\Cdtmsserver.key"

# This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
# openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh dh1024.pem

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
#Launch VPN server on TAP/TUN interface with Specific IP

```

address and net mask

**;server 192.168.0.0 255.255.255.0**

# Maintain a record of client <-> virtual IP address  
 # associations in this file. If OpenVPN goes down or  
 # is restarted, reconnecting clients can be assigned  
 # the same virtual IP address from the pool that was  
 # previously assigned.

**ifconfig-pool-persist ipp.txt**

# Configure server mode for ethernet bridging.  
 # You must first use your OS's bridging capability  
 # to bridge the TAP interface with the ethernet  
 # NIC interface. Then you must manually set the  
 # IP/netmask on the bridge interface, here we  
 # assume 10.8.0.4/255.255.255.0. Finally we  
 # must set aside an IP range in this subnet  
 # (start=10.8.0.50 end=10.8.0.100) to allocate  
 # to connecting clients. Leave this line commented  
 # out unless you are ethernet bridging.

**#If users want to launch the VPN server on bridge  
 server-bridge 10.8.0.2 255.255.255.0 10.8.0.50 10.8.0.100**

# Configure server mode for ethernet bridging  
 # using a DHCP-proxy, where clients talk  
 # to the OpenVPN server-side DHCP server  
 # to receive their IP address allocation  
 # and DNS server addresses. You must first use  
 # your OS's bridging capability to bridge the TAP  
 # interface with the ethernet NIC interface.  
 # Note: this mode only works on clients (such as  
 # Windows), where the client-side TAP adapter is  
 # bound to a DHCP client.

**;server-bridge**

# Push routes to the client to allow it  
 # to reach other private subnets behind  
 # the server. Remember that these  
 # private subnets will also need  
 # to know to route the OpenVPN client

```

# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
;push "route 192.168.35.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"

# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
client-config-dir ccd
;route 192.168.40.128 255.255.255.248

# Then create a file ccd/Thelonious with this line:
# iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN. This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
# ifconfig-push 10.9.0.1 10.9.0.2
# Suppose that you want to enable different
# firewall access policies for different groups
# of clients. There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
# group, and firewall the TUN/TAP interface
# for each group/daemon appropriately.

```

```
# (2) (Advanced) Create a script to dynamically
# modify the firewall in response to access
# from different clients. See man
# page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
;push "redirect-gateway def1 bypass-dhcp"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses. CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by.opendns.com.
;push "dhcp-option DNS 10.10.22.243"
;push "dhcp-option WINS 202.106.0.20"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
client-to-client

# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names. This is recommended
# only for testing purposes. For production use,
# each client should have its own certificate/key
```

```
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
```

### **duplicate-cn**

```
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
#( Openvpn can not connect again in mode server )
# The value can be modified by users.
```

### **keepalive 10 120**

```
# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
# openvpn --genkey --secret ta.key
```

```
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret
```

```
# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
#We use DES-CBC as an example.
;cipher BF-CBC
;cipher AES-128-CBC
;cipher DES-EDE3-CBC
```

### **cipher DES-CBC**

```
# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
;comp-lzo no

# The maximum number of concurrently connected
# clients we want to allow.
# The value can be modified by users.
max-clients 20

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nobody

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun

# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it. Use one
```



```

# or the other (but not both).
;log openvpn.log
;log-append openvpn.log

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
#The value can be modified by users
verb 4

# Silence repeating messages. At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20

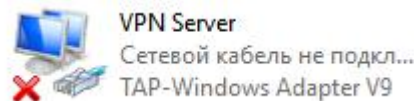
```

**Примечание:** комментарии отмечены «#» или «;». Данная конфигурация работает для ОС Windows или Linux.

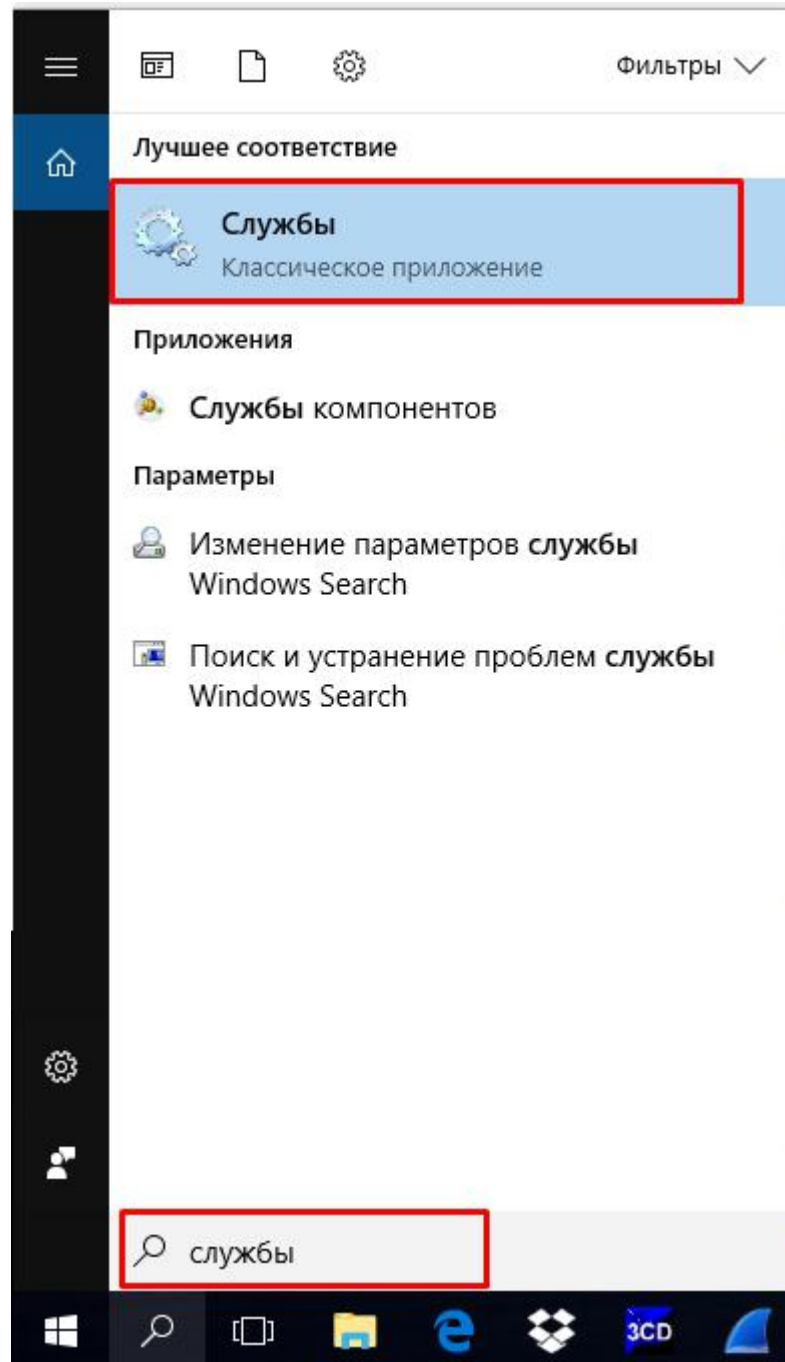
Помните, что в ОС Windows для задания пути, необходимо использовать «\|» (например, "C:\|Program Files\|OpenVPN\|config\|foo.key")

### 2.3.2 Подключение к серверу OpenVPN.

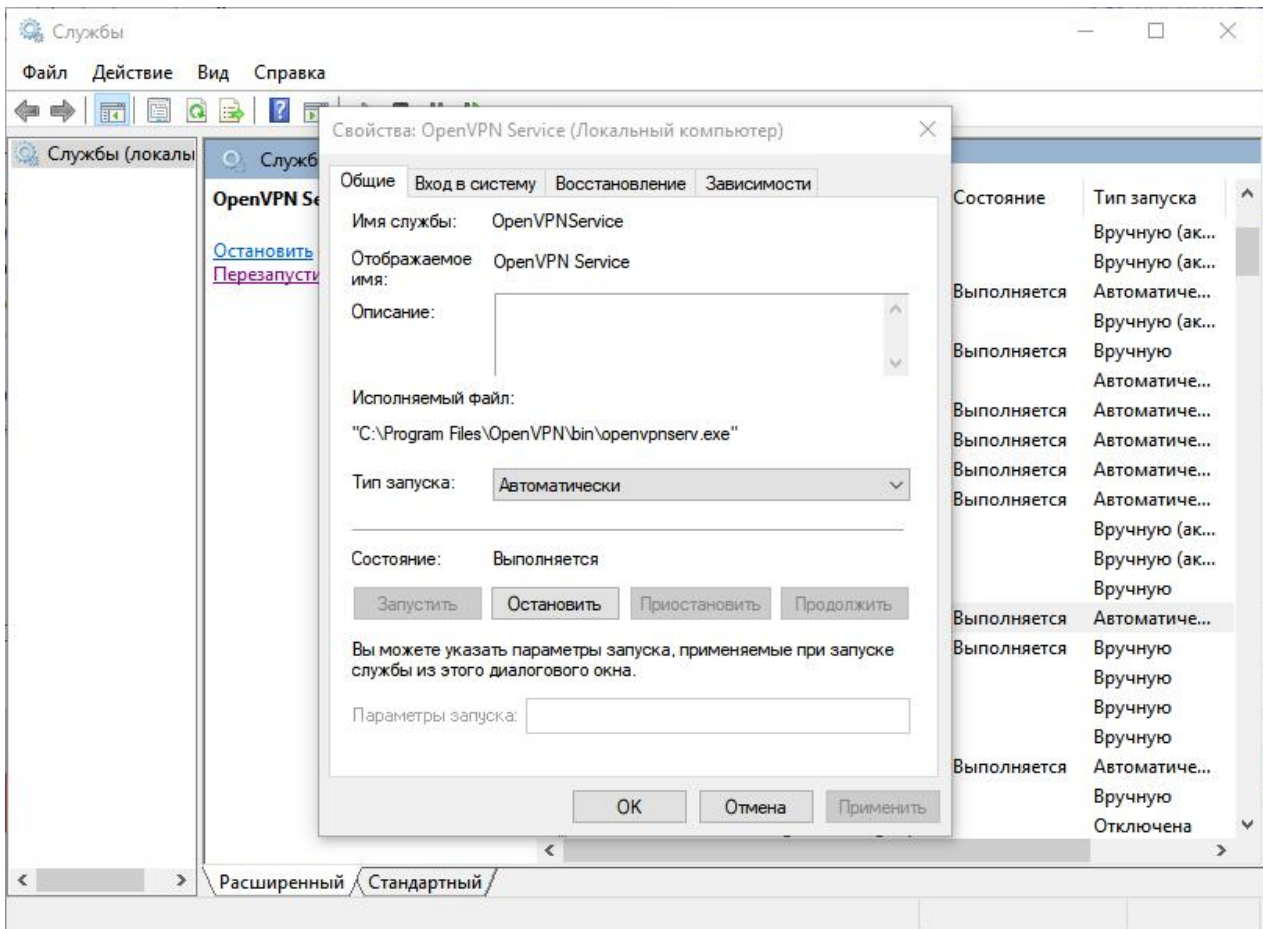
- В сетевых подключениях Windows открываем управление адаптерами - TAP-адаптер переименовываем в «VPN Server» (как у нас указано в конфигурационном файле, разделе dev-node):



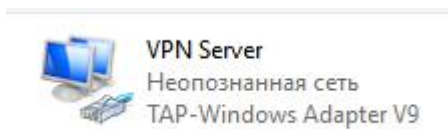
- Открываем службы Windows и находим «OpenVpnService».



- Службу «OpenVpnService» настраиваем на автозапуск и включаем:



- Ранее переименованный сетевой интерфейс должен включиться:



### 2.3.3 Конфигурация клиента.

Переходим в папку **C:\Program Files\OpenVPN\sample-config**.  
Открываем и редактируем файл **client.ovpn**.

```
#####
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server. #
# #
# This configuration can be used by multiple #
# clients, however each client should have #
# its own cert and key files. #
# #
```

```

# On Windows, you might want to rename this #
# file so it has a .ovpn extension #
#####
# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
Client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
#we use Tap as an example
dev tap
;dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one. On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap
# Are we connecting to a TCP or
# UDP server? Use the same setting as
# on the server.
#Please be same as the server's protocol.
;proto tcp
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
#Outer network ip of OpenVPN
remote 192.168.35.91 1194
;remote my-server-2 1194

# Choose a random host from the remote

```

```
# list for load-balancing. Otherwise
# try hosts in the order specified.
;remote-random

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server. Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
Nobind

# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nobody
# Try to preserve some state across restarts.
persist-key
persist-tun

# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here. See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets. Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings

# SSL/TLS parms.
```

```
# See the server config file for more
# description. It's best to use
# a separate .crt/.key file pair
# for each client. A single ca
# file can be used for all clients.
#Please be sure the file name
ca /config/openvpn/ca.crt
cert /config/openvpn/Client.crt
key /config/openvpn/Client.key

# Verify server certificate by checking
# that the certificate has the nsCertType
# field set to "server". This is an
# important precaution to protect against
# a potential attack discussed here:
# http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the nsCertType
# field set to "server". The build-key-server
# script in the easy-rsa folder will do this.
ns-cert-type server

# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
;cipher x
# we use DES-CBC as an example
cipher DES-CBC

# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
;comp-lzo no
```

```
# Set log file verbosity.
verb 3

# Silence repeating messages
;mute 20
```

### 2.3.4 Создание файла **client.tar**

Для создание клиентского файла необходимо:

- Скопировать файл **client.ovpn** из **C:\Program Files\OpenVPN\sample-config** и **Client.crt**, **Client.key**, **ca.crt** файлов **C:\Program Files\OpenVPN\easy-rsa\** и **C:\Program Files\OpenVPN\** для клиента.
- Переименовать файл **client.ovpn** в **vpn.conf**.
- Файл **vpn.conf** привести к следующему виду, соблюдая весь синтаксис:

```
client
resolv-retry infinite
nobind
remote 172.31.71.94 12345
proto udp
dev tun
comp-lzo
ca /config/openvpn/ca.crt
cert /config/openvpn/clients.crt
key /config/openvpn/clients.key
dh /config/openvpn/dh1024.pem
float
cipher DES-CBC
keepalive 10 120
persist-key
persist-tun
verb 0
```

- При помощи программы 7-zip для создать файл **vpn.tar**.

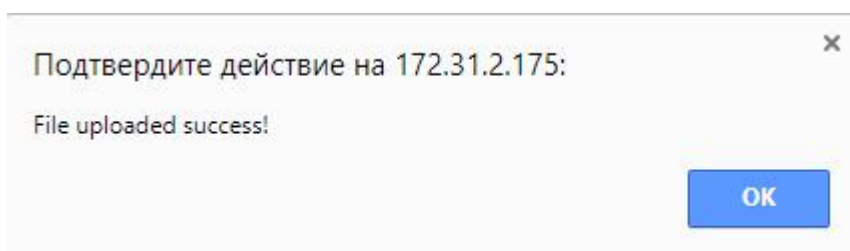
### 3. Настройка телефона


#### 3.1 Конфигурация через web

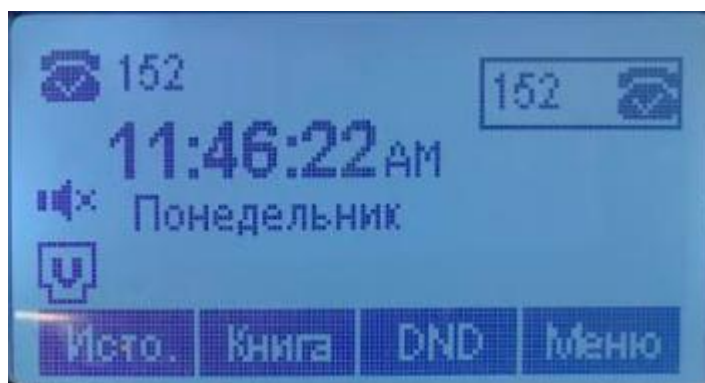
В разделе **Network-Advanced-VPN**:

- Устанавливаем маркер напротив функции **OpenVPN**
- Включаем данный функционал

- Загружаем созданный файл **vpn.tar**
- Нажимаем клавишу **«Upload»**
- После успешной загрузки загрузки появится сообщение:



- Активируем **OpenVPN** и нажимаем кнопку **«Submit»**
- После перезагрузки аппарат применит настройки и при успешном подключении VPN-сервера на дисплее отобразится значок 





Через экранной меню можно увидеть присвоенный ip адрес.

