

OPNENVPN – SETUP AND CONFIGURATION

DOCUMENTATION

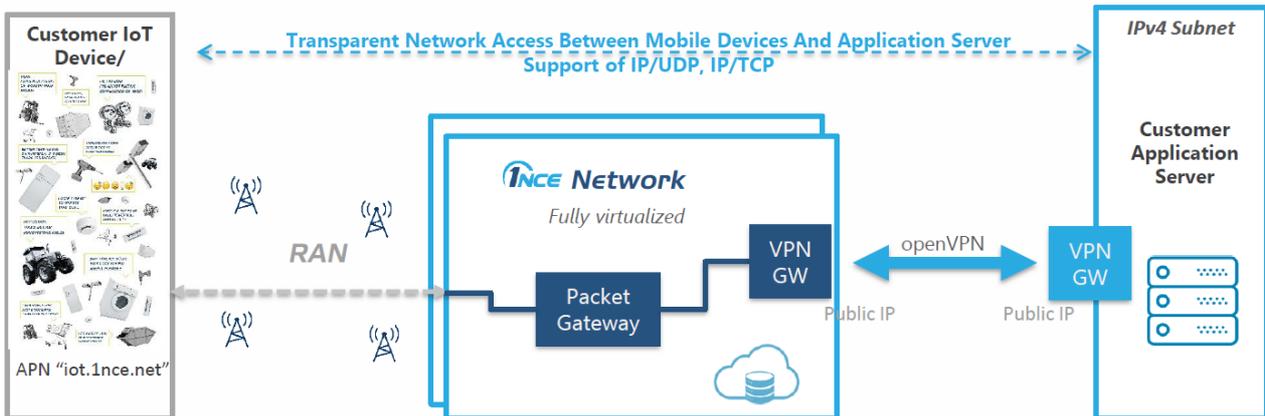
Version 2.0 as of February 2019

Table of Contents

1	General Information.....	2
2	Installing OpenVPN in Windows	3
3	Installing OpenVPN in Linus/MacOS	4
4	Example Setup with two Raspberrys and connection test on Linux.....	4
4.1	raspberrypi2:	4
4.2	raspberrypi3:	5

1 GENERAL INFORMATION

Open VPN is the recommended application setup by 1NCE to establish a secure data connection between the 1NCE network and the customers server.



Therefore, the OpenVPN client needs to be installed on the customer server (aka application server) to which the data of the device shall be transmitted. The OpenVPN connection to the 1NCE network is established by the OpenVPN client, that can be downloaded via the 1NCE customer portal. There are two different versions for Windows and for Linux/MacOS available.

After the network connection and the OpenVPN connections have been established, you may check the connection with a Ping:

SIM-> Server

From the device execute a PING to the OpenVPN client's address (e.g. 10.x.x.x; the IP is always the same for the customer).

Server-> SIM

From your server execute a PING to the IP address of your device. It is a private IPv4 address which has been assigned by the 1NCE network when the mobile connection is established (e.g. 100.x.x.x; the IP address is always the same for one SIM).

2 INSTALLING OPENVPN IN WINDOWS

Install OpenVPN software from <https://openvpn.net/index.php/open-source/downloads.html>

Source Tarball (gzip)	openvpn-2.4.6.tar.gz	GnuPG Signature
Source Tarball (xz)	openvpn-2.4.6.tar.xz	GnuPG Signature
Source Zip	openvpn-2.4.6.zip	GnuPG Signature
Installer, Windows 7 and later	openvpn-install-2.4.6-I602.exe	GnuPG Signature

NOTE: the GPG key used to sign the release files has been changed since previous releases. The new GPG public key and signatures are available [here](#).

We also provide static URLs pointing to latest releases to ease automation.

This release is also available in our own software repositories for Debian and amd64. For details, look [here](#).

You can use [EasyRSA 2](#) or [EasyRSA 3](#) for generating your own certificates and installers. The latter is a more modern alternative for UNIX-like operating systems.

The Windows installers are bundled with OpenVPN-GUI - its source code is available on an [alternative download server](#).

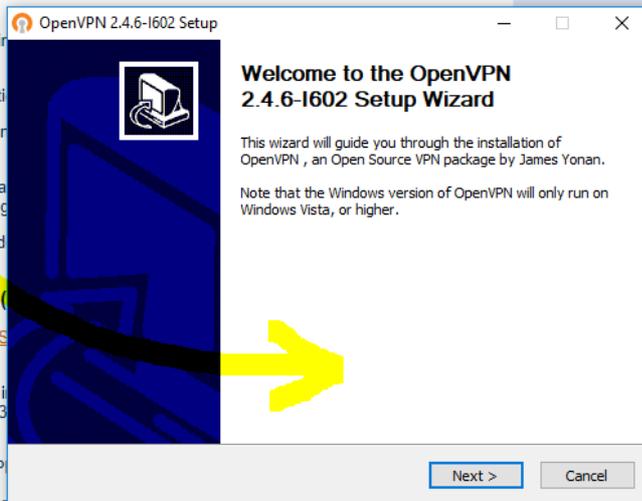
OpenVPN 2.3.18 (old stable) -- released on 2017.09.26 (2017.09.26)

This is a minor release. It fixes the [key-method 1](#) problem as well the [NSI problems](#) described in more detail in OpenVPN 2.4.5 release notes.

Windows installers I602 and I002 fix [Trac issue #948](#) which caused the installer to be 100 characters long. In addition easy-rsa has been updated from 2.2.0 to 2.3.0 to fix the directory being in the system PATH.

NOTE: The GPG key used to sign release files [has changed](#) in OpenVPN 2.4.5.

This release is the latest *old stable* release, and the last major release to support Windows 7. Normally you should use the latest stable release (2.4.x) instead.



Download [client.ovpn](#) from "Configuration" section in the customer portal and store it on your server in C:\Program Files\OpenVPN\config

Download the file [credentials.txt](#) from "Configuration" section in the customer portal and save it in "C:\Program Files\OpenVPN\config"

Start OpenVPN GUI application and monitor the connection. For a log on your last connection open the logfile in C:\Program Files\OpenVPN\log\client.txt

(may also be C:\Users\<IHRPROFIL>\OpenVPN\log\client.txt)

3 INSTALLING OPENVPN IN LINUS/MACOS

Install OpenVPN with `sudo apt-get install openvpn`

Download „client.conf“ from customer portal (Configuration -> OpenVPN Configuration -> Linux/macOS -> Download client.conf).

* Optional: Added line `log /var/log/openvpn_1nce.log` just for debugging purpose. *

Copy <client.conf> to /etc/openvpn.

Download „credentials.txt“ from customer portal (Configuration -> OpenVPN Configuration -> Linux/macOS -> 3. Save the following file credentials.txt ...).

Copy <credentials.txt> also in /etc/openvpn.

Start the OpenVPN connection with new configuration.

```
sudo service openvpn restart
```

4 EXAMPLE SETUP WITH TWO RASPBERRYS AND CONNECTION TEST ON LINUX

The test environment shows how to check if the OpenVPN connection has successfully been established.

The following test environments have been run on a Linux server.:

Two Raspberry Pis have been used in this test environment. They are called „raspberrypi2“ and „raspberrypi3“.

The following steps show how to check that the OpenVPN connection has been successfully established:

4.1 RASPBERRYPI2:

The raspberrypi2 as data receiving application server establishes the OpenVPN connection.

Look into logfile, e.g. with `<sudo cat /var/log/openvpn_1nce.log>`.

Here are the last lines of the test:

```
Mon Aug 27 15:31:45 2018 TUN/TAP device tun0 opened
Mon Aug 27 15:31:45 2018 TUN/TAP TX queue length set to 100
Mon Aug 27 15:31:45 2018 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Mon Aug 27 15:31:45 2018 /sbin/ifconfig tun0 10.64.71.XX pointopoint 10.64.71.XX mtu 1500
Mon Aug 27 15:31:45 2018 /sbin/route add -net 10.64.X.X netmask 255.255.255.255 gw 10.64.71.XX
Mon Aug 27 15:31:45 2018 /sbin/route add -net 100.117.XXX.0 netmask 255.255.252.0 gw 10.64.71.XX
Mon Aug 27 15:31:45 2018 GID set to nogroup
Mon Aug 27 15:31:45 2018 UID set to root
Mon Aug 27 15:31:45 2018 Initialization Sequence Completed
```

In this example 10.64.71.XX is the address of the OpenVPN client.

Type in <ifconfig> and search for tun0. In the test we have:

```
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.64.71.XX  P-t-P:10.64.71.XX  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1344 (1.3 KiB)  TX bytes:1428 (1.3 KiB)
```

4.2 RASPBERRYPI3:

The raspberry3, in this case a 3G USB Surfstick, Huawei E173, is in this test environment the SIM carrying device (modem).

Using wvdial with following config to get the default configuration of your device:

```
$ cat /etc/wvdial.conf

[Dialer Ince]
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2
Init3 = AT+CGDCONT=1,"IP","iot.ince.net"
Modem Type = Analog Modem
Baud = 9600
```

```
New PPPD = yes
Modem = /dev/ttyUSB1
ISDN = 0
Phone = *99#
Password = *
Username = *
```

Please note that this config is based on this test environment. If you have a different modem you have to run

```
sudo wvdialconf
```

first, then set the 1NCE APN and save this configuration with a name of your choice. In this test environment we have named the config file with "1nce".

Starting the connection with

```
$ sudo wvdial 1nce > /tmp/3G_1nce.log 2>&1
```

Looking into the logfile you see that the connection is established.

```
$ cat /tmp/3G_1nce.log
--> WvDial: Internet dialer version 1.61
--> Initializing modem.
--> Sending: ATZ
ATZ
OK
--> Sending: ATQ0 V1 E1 S0=0 &C1 &D2
ATQ0 V1 E1 S0=0 &C1 &D2
OK
--> Sending: AT+CGDCONT=1,"IP","iot.1nce.net"
AT+CGDCONT=1,"IP","iot.1nce.net"
OK
--> Modem initialized.
--> Sending: ATDT*99#
--> Waiting for carrier.
ATDT*99#
CONNECT
--> Carrier detected. Waiting for prompt.
--> Don't know what to do! Starting pppd and hoping for the best.
--> Starting pppd at Mon Aug 27 15:35:54 2018
```

```
--> Pid of pppd: 12919
--> Using interface ppp0
--> pppd: [1a]
--> local IP address 100.117.XXX.5
--> pppd: [1a]
--> remote IP address 10.64.64.XX
--> pppd: [1a]
--> primary DNS address 8.8.8.8
--> pppd: [1a]
--> secondary DNS address 8.8.4.4
--> pppd: [1a]
```

The private IP address in this case is 100.117.XXX.5, which is the IP of the device.

Now you have a direct connection through the OpenVPN tunnel to the raspberrypi2, which is the application server in this test environment.

With a ping you can try it out:

```
raspberrypi3 $ ping -I ppp0 10.64.71.XX
PING 10.64.71.57 (10.64.71.57) from 100.117.XXX.5 ppp0: 56(84) bytes of data.
64 bytes from 10.64.71.XX: icmp_seq=1 ttl=62 time=734 ms
64 bytes from 10.64.71.XX: icmp_seq=2 ttl=62 time=518 ms
64 bytes from 10.64.71.XX: icmp_seq=3 ttl=62 time=118 ms
64 bytes from 10.64.71.XX: icmp_seq=4 ttl=62 time=116 ms
64 bytes from 10.64.71.XX: icmp_seq=5 ttl=62 time=95.9 ms
64 bytes from 10.64.71.XX: icmp_seq=6 ttl=62 time=94.8 ms
64 bytes from 10.64.71.XX: icmp_seq=7 ttl=62 time=103 ms
64 bytes from 10.64.71.XX: icmp_seq=8 ttl=62 time=102 ms
64 bytes from 10.64.71.XX: icmp_seq=9 ttl=62 time=101 ms
64 bytes from 10.64.71.XX: icmp_seq=10 ttl=62 time=100 ms
64 bytes from 10.64.71.XX: icmp_seq=11 ttl=62 time=99.9 ms
^C
--- 10.64.71.XX ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10008ms
rtt min/avg/max/mdev = 94.893/198.913/734.189/206.778 ms
```

```
raspberrypi2 $ ping -I tun0 100.117.XXX.5
PING 100.117.224.5 (100.117.224.5) from 10.64.71.57 tun0: 56(84) bytes of data.
64 bytes from 100.117.XXX.5: icmp_req=1 ttl=62 time=804 ms
64 bytes from 100.117.XXX.5: icmp_req=2 ttl=62 time=133 ms
64 bytes from 100.117.XXX.5: icmp_req=3 ttl=62 time=102 ms
64 bytes from 100.117.XXX.5: icmp_req=4 ttl=62 time=101 ms
64 bytes from 100.117.XXX.5: icmp_req=5 ttl=62 time=100 ms
64 bytes from 100.117.XXX.5: icmp_req=6 ttl=62 time=121 ms
64 bytes from 100.117.XXX.5: icmp_req=7 ttl=62 time=99.9 ms
64 bytes from 100.117.XXX.5: icmp_req=8 ttl=62 time=99.5 ms
64 bytes from 100.117.XXX.5: icmp_req=9 ttl=62 time=99.1 ms
64 bytes from 100.117.XXX.5: icmp_req=10 ttl=62 time=97.4 ms
64 bytes from 100.117.XXX.5: icmp_req=11 ttl=62 time=107 ms
^C
--- 100.117.XXX.5 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10007ms
rtt min/avg/max/mdev = 97.494/169.764/804.094/200.879 ms
```

After having established and tested the OpenVPN connection you may terminate the connection with:

```
$ sudo killall wvdial
$ tail -f /tmp/3G_lnce.log
Caught signal 15: Attempting to exit gracefully...
--> Terminating on signal 15
--> pppd: [1a]
--> Connect time 108.0 minutes.
--> pppd: [1a]
--> pppd: [1a]
--> pppd: [1a]
--> pppd: [1a]
--> Disconnecting at Mon Aug 27 17:23:53 2018
```