OpenVPN Toolkit for FP Gateways
Quick Guide

Version: 1.3.1

www.inovolabs.com

Publication date: 23/02/2021

Table of contents

1 Introduction

FP IIoT gateways as of the 6th generation (FP S-ENGuard, FP S-OTGuard, FP S-Compact) support establishing a VPN tunnel as of firmware version 5.2.0.2.

In order to configure a VPN tunnel for an FP gateway, the following steps are required in principle:

1. Obtain the OpenVPN configuration file (`*.ovpn`) and certificates / keys
2. Generate the TiXML configuration and transfer it to the FP gateway
3. Restart the device

Unpack the OpenVPN toolkit in a new directory.
This generates the following subdirectories:

Directory	**Explanation**
SetBinary	OpenVPN files as XML image
config	Contains the OVPN configuration files/certificates, etc.
TICO	TiXML device configuration to establish the tunnel

2 VPN configuration files

A VPN configuration file generally comprises a control file (`*.ovpn`), certificates and keys to establish a secure connection.
An optional password file can also be defined.

The certificates can either be provided as single files (these are then linked via entries in the `*.ovpn` file) or integrated directly into the `*.ovpn` file.

Check whether the "`nobind`" option is used in the `*.ovpn` configuration file.
If this option is use actively, please comment out this option with the # character:

```
# nobind
```

In order to transfer the VPN configuration files to the FP gateway, the files must first be transformed into XML format. The websrc.exe FP Windows command line tool is used to do this.

- Copy all required VPN configuration files to the "`config`" subdirectory.
- Start the "`make_vpn_config.bat`" file in a Windows prompt.

The TiXML version of the VPN configuration files was generated in the "`SetBinary`" directory.
The file is called "`40-VPNconf.txt`".

- Use the TICO software to transfer the "`40-VPNconf.txt`" file to the FP gateway

3 TiXML configuration

In order that the FP gateway actually uses the VPN configuration, you must adjust a TiXML configuration and then transfer it to the device.

a) VPN basic configuration

File: `TICO\ISP_VPN_Config.txt`

```
[<SetConfig _="ISP">
  <VPN>
     <Config _="NameOfConfigurationFile.ovpn"/>
     <SignalOK _="9"/>
    <SignalError _="2"/>
  </VPN>
</SetConfig>]
```

Replace the text "NameOfConfigurationFile.ovpn" with the real name of your VPN configuration file (e.g. `vpnclient.ovpn`).

Leave the settings for `SignalOK` and `SignalError` as they are at first.
These settings are used to use the signal LED to display the VPN tunnel's status:
flashing red = connection being established; illuminated green = tunnel established.

- Change the VPN configuration shown above according to your requirements.
- Save your changes.
- Use the TICO software to transfer the "`ISP_VPN_Config.txt`" file to the device.

b) Optional: VPN routing

File: `TICO\ISP_VPN_Routing.txt`

The `ISP/ISP/OUT` database can be used to define the WAN interface via which the VPN tunnel is to be established. The routing is defined separately for this.
The `<VPN _="..."/>` entry is relevant for the VPN tunnel.

```
[<SetConfig _="ISP/ISP">
<OUT>
  <SMTP _="MODEM" />
  <CBIS _="MODEM" />
  <POP3 _="MODEM" />
  <URLSend _="MODEM" />
  <INetTime _="MODEM" />
  <HTTPConn _="MODEM" />
  <CloudConn _="MODEM" />
  <IBMConn _="MODEM" />
  <FTPPut _="MODEM" />
  <SFTPPut _="MODEM" />
     <!-- VPN tunnel is established via LAN (alternative: MODEM) -->
  <VPN _="Ethernet" />
</OUT>
</SetConfig>]
```

Possible routes are:
"Ethernet" = LAN interface
"MODEM" = GPRS / UMTS / LTE

Ensure that the entries are used exactly as specified above
(the system distinguishes between uppercase and lowercase!).

- Change the VPN configuration shown above according to your requirements.
- Save your changes.
- Use the TICO software to transfer the "ISP_VPN_Routing.txt" file to the device.

4 Testing the configuration

Restart the FP gateway.
After the device has issued an audible signal, it is ready for operation.
Around 30-60 seconds after the audible signal, the “Signal” LED should flash red.
This indicates that the VPN tunnel is starting to be established.

The “Signal” LED should illuminate green within a few seconds.
The VPN tunnel is therefore established successfully.

Setting up an additional virtual IP address

If you wish to communicate with the device via the VPN tunnel using a specific IP address, you can assign the FP gateway’s LAN interface a second IP address. This is assigned in the ISP/Ethernet database.

Example:

```
[<SetConfig _="ISP">
  <Ethernet>
    <IP _="DHCP"/>

    <!-- second (alias) IP address 10.0.0.1 -->
    <IP_2 _="10.0.0.1"/>
    <Mask_2 _="255.255.255.0"/>

  </Ethernet>
</SetConfig>]
```

The configuration shown above uses DHCP on the LAN interface.
Address "10.0.0.1" is configured as a second (alias) IP address together with the network mask "255.255.255.0".
The alias IP address can be used to access the device via the tunnel from externally. The internal Linux device designation for the alias IP address is "eth0:1".

5 Error diagnostics

If no VPN tunnel is established, you should test the configuration on the Linux command line. To do this, start the telnet service on the FP gateway via TICO by entering the following command in test mode:

```
[<LinuxCmd _="telnetd" mode="start" magic="03040608890"/>]
```

The telnet daemon should be started after one minute at the most.
Then use a terminal program to connect to the FP gateway on port 23.

Login: root
Password: HTBasic

Now enter the following commands (Linux prompt # is also displayed):

```
# tdg kill
# killall openvpn
# cd /flash0/app/VPN
# openvpn --config NameOfConfigurationFile.ovpn
```

Replace "NameOfConfigurationFile.ovpn" with the real name of your configuration file again (e.g "`vpntest.ovpn`").

You can now track how the tunnel is established directly in Linus and see any errors in the command line.

In order to check whether the tunnel was established, call the Linux tool "`ifconfig`" at Linux level and check whether an interface with the name "`tun0`" exists and whether an IP address was assigned:

```
# ifconfig
```

Result (example):

```
eth0      Link encap:Ethernet  HWaddr 00:11:E8:25:16:A6
          inet addr:192.168.167.232  Bcast:192.168.167.255  Mask:255.255.255.128
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:331303 errors:0 dropped:123 overruns:0 frame:0
          TX packets:224031 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:30731832 (29.3 MiB)  TX bytes:41277967 (39.3 MiB)
          Interrupt:24 Base address:0xc000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:27435 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27435 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2398089 (2.2 MiB)  TX bytes:2398089 (2.2 MiB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.6  P-t-P:10.8.0.5  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```