OpenVPN Toolkit für FP Gateways

Kurzdokumentation

Version: 1.3.0

www.inovolabs.com

Redaktionsschluss: 31.07.2020

Inhaltsverzeichnis

1 EINFÜHRUNG 3

2 VPN KONFIGURATIONSDATEIEN 3

3 TIXML-KONFIGURATION 4

4 TEST DER KONFIGURATION 5

5 FEHLERDIAGNOSE 6

1 Einführung

FP IIoT Gateways ab der 6.Generation (FP S-ENGuard, FP S-OTGuard, FP S-Compact) unterstützen ab Firmware-Version 5.2.0.2 den Aufbau eines VPN-Tunnels.

Um einen VPN-Tunnel für ein FP-Gateway zu konfigurieren, sind prinzipiell folgende Schritte notwendig:

1. OpenVPN Konfigurationsdatei (`*.ovpn`) und Zertifikate / Schlüssel beschaffen
2. TiXML-Konfiguration erzeugen und in das FP-Gateway übertragen
3. Gerät neu starten

Entpacken Sie das OpenVPN Toolkit in ein neues Verzeichnis.
Folgende Unterverzeichnisse werden dabei erzeugt:

Verzeichnis	**Erläuterung**
SetBinary	OpenVPN-Dateien als XML-Image
config	Enthält die OVPN Konfigurationsdateien/Zertifikate etc.
TICO	TiXML Gerätekonfiguration für den Aufbau des Tunnels

2 VPN Konfigurationsdateien

Eine VPN-Konfigurationsdatei besteht in der Regel aus einer Steuerdatei (`*.ovpn`), Zertifikaten und Schlüsseln für den Aufbau einer gesicherten Verbindung.
Optional läßt sich auch eine Passwortdatei definieren.

Die Zertifikate können entweder als Einzeldatei bereitgestellt werden (diese sind dann über Einträge in der `*.ovpn`-Datei verlinkt) oder direkt in die `*.ovpn`-Datei integriert werden.

Prüfen Sie, ob in der Konfigurationsdatei `*.ovpn` die Option "`nobind`" verwendet wird.
Wenn diese Option aktiv verwendet wird, kommentieren Sie diese Option bitte mit dem Zeichen # aus:

```
# nobind
```

Um die VPN-Konfigurationsdateien in das FP-Gateway zu übertragen, müssen die Dateien zunächst in ein XML-Format transformiert werden. Dazu wird das FP Windows-Kommandozeilentool websrc.exe verwendet.

- Kopieren Sie alle benötigten VPN-Konfigurationsdateien in das Unterverzeichnis "`config`".
- Starten Sie die Datei "`make_vpn_config.bat`" in einer Windows-Eingabeaufforderung

Im Verzeichnis "`SetBinary`" wurde nun die TiXML-Version der VPN-Konfigurationsdateien erzeugt. Die Datei heißt "`40-VPNconf.txt`".

- Übertragen Sie die Datei "`40-VPNconf.txt`" mit der Software TICO in das FP-Gateway

3 TiXML-Konfiguration

Damit das FP-Gateway die VPN-Konfiguration tatsächlich verwendet, müssen Sie eine TiXML-Konfiguration anpassen und dann in das Gerät übertragen.

a) VPN Basiskonfiguration

Datei: `TICO\ISP_VPN_Config.txt`

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

Ersetzen Sie den Text "NameDerKonfigurationsdatei.ovpn" mit dem tatsächlichen Namen Ihrer VPN-Konfiguratonsdatei (z.B. `vpnclient.ovpn`).

Die Einstellungen für `SignalOK` und `SignalError` sollten Sie zunächst so belassen. Mit diesen Einstellungen wird über die Signal-LED der Zustand des VPN-Tunnels angezeigt: rot blinkend = Verbindungsaufbau; grün leuchtend = Tunnel aufgebaut

- Ändern Sie die oben gezeigte Konfiguration für VPN nach Ihren Anforderungen
- Speichern Sie die Änderungen
- Übertragen Sie die Datei "`ISP_VPN_Config.txt`" mit der Software TICO in das Gerät

b) Optional: VPN Routing

Datei: `TICO\ISP_VPN_Routing.txt`

Über die Datenbank `ISP/ISP/OUT` kann bestimmt werden, über welche WAN-Schnittstelle der VPN-Tunnel aufgebaut werden soll. Das Routing wird für jeden Dienst getrennt festgelegt. Für den VPN-Tunnel ist der Eintrag `<VPN _="..."/>` relevant.

```
[<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 wird ueber LAN aufgebaut (Alternative: MODEM) -->
 <VPN _="Ethernet" />
</OUT>
</SetConfig>]
```

Mögliche Routen sind:
"Ethernet" = LAN-Schnittstelle
"MODEM" = GPRS / UMTS / LTE

Bitte achten Sie darauf, dass die Einträge exakt wie oben angegeben verwendet werden (Gross- / Kleinschreibung wird unterschieden!).

- Ändern Sie die oben gezeigte Konfiguration für VPN nach Ihren Anforderungen
- Speichern Sie die Änderungen
- Übertragen Sie die Datei "ISP_VPN_Routing.txt" mit der Software TICO in das Gerät

4 Test der Konfiguration

Starten Sie das FP-Gateway neu.
Nachdem das Gerät ein akustisches Signal ausgegeben hat, ist es betriebsbereit.
Ca. 30..60 Sekunden nach dem akustischen Signal sollte die LED "Signal" rot blinken.
Dies signalisiert den Beginn des Aufbaus des VPN-Tunnels.

Nach wenigen Sekunden sollte die "Signal"-LED dauerhaft grün leuchten.
Damit ist der VPN-Tunnel erfolgreich aufgebaut.

Zusätzliche virtuelle IP-Adresse einrichten

Will man über den VPN-Tunnel mit einer bestimmten IP-Adresse auf dem Gerät kommunizieren, kann man dem LAN-Interface des FP-Gateways eine zweite IP-Adresse zuweisen. Die Zuweisung erfolgt in der Datenbank ISP/Ethernet.

Beispiel:

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

    <!-- zweite (Alias-) IP-Adresse 10.0.0.1 -->
    <IP_2 _="10.0.0.1"/>
    <Mask_2 _="255.255.255.0"/>

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

Die oben gezeigte Konfiguration verwendet DHCP auf dem LAN-Interface.
Als zweite (Alias-) IP-Adresse wird die Adresse "10.0.0.1" konfiguriert, zusammen mit der Netzmaske "255.255.255.0".
Die Alias-IP-Adresse kann verwendet werden, um von außen über den Tunnel auf das Gerät zuzugreifen. Die interne Linux-Gerätebezeichnung für die Alias-IP-Adresse lautet "eth0:1".

5 Fehlerdiagnose

Sollte kein VPN-Tunnel aufgebaut werden, sollte man die Konfiguration auf der Linux-Kommandozeile testen. Dazu startet man den telnet-Dienst auf dem FP-Gateway über die TICO, in dem man im Testmodus den Befehl eingibt:

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

Nach spätestens einer Minute sollte der telnet-Daemon gestartet sein.
Verbinden Sie sich dann über ein Terminalprogramm mit dem FP-Gateway auf Port 23.

Login: root
Password: HTBasic

Geben Sie nun die folgenden Befehle ein (Linux-Prompt # wird mit angezeigt):

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

Ersetzen Sie hierbei wiederum "NameDerKonfigurationsdatei.ovpn" mit dem realen Namen Ihrer Konfigurationsdatei (z.B. "`vpntest.ovpn`").

Sie können nun direkt im Linux verfolgen, wie der Tunnel aufgebaut wird und sehen etwaige Fehler auf der Kommandozeile.

Um zu überprüfen, ob der Tunnel aufgebaut wurde, rufen Sie das Linux-Tool "`ifconfig`" auf der Linux-Ebene auf und überprüfen Sie, ob ein Interface mit dem Namen "`tun0`" existiert und eine IP-Adresse zugewiesen wurde:

```
# ifconfig
```

Ergebnis (Beispiel):

```
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)
```