OPENVPN – SETUP UND KONFIGURATION

DOKUMENTATION

Version 2.0, Stand Februar 2019

Inhaltsverzeichnis

1 Allgemeine Informationen 2

2 Installation von OpenVPN auf Windows 3

3 Installation von OpenVPN auf Linux/MacOS 4

4 Beispiel eines Setup mit zwei Raspberrys und Verbindungstest auf Linux 4

4.1 raspberrypi2: 4

4.2 raspberrypi3: 5

5 Fehlerbehebung 9

5.1 Lokalisierung der Logdateien 9

5.2 Routingtabelle für OpenVPN 10

5.3 Potenzieller route-subnetz-Konflikt 10

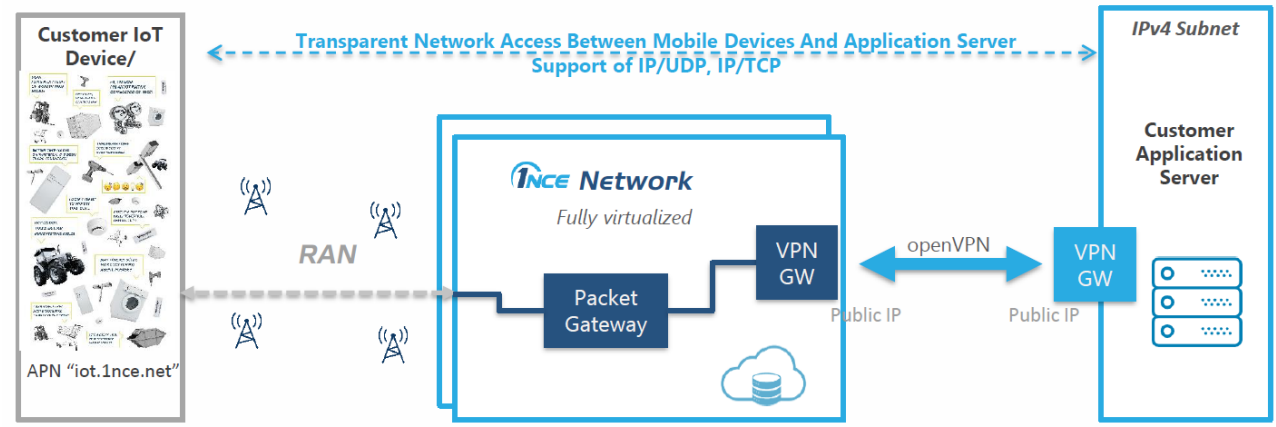
5.4 OpenVPN Verbindung ohne Grund abgebrochen 10

5.5 HTTPs Timeout durch OpenVPN Verbindung 11

5.6 OpenVPN Authentifizierungsfehler 11

1 ALLGEMEINE INFORMATIONEN

OpenVPN ist die von 1NCE empfohlene Anwendung, um eine sichere Datenverbindung zwischen dem 1NCE-Netzwerk und dem Kundenserver herzustellen.



Der OpenVPN-Client muss auf dem Kundenserver (auch bekannt als Applikationsserver) installiert sein, an den die Daten des Gerätes übertragen werden sollen. Die OpenVPN-Verbindung zum 1NCE-Netzwerk wird vom OpenVPN-Client hergestellt, der über das 1NCE-Kundenportal heruntergeladen werden kann. Es gibt zwei verschiedene Versionen des Clients, zum einen für Windows und zum anderen für Linux/MacOS.

Nachdem die Netzwerkverbindung und die OpenVPN-Verbindung hergestellt wurden, können Sie die Verbindung mit einem Ping überprüfen:

SIM-> Server

Führen Sie vom Gerät aus einen PING zur Adresse des OpenVPN-Clients aus (z.B. 10.x.x.x.x.x; die IP ist für den Kunden immer die gleiche).

Server-> SIM

Führen Sie auf Ihrem Server einen PING zur IP-Adresse Ihres Geräts aus. Es handelt sich um eine private IPv4-Adresse, die vom 1NCE-Netzwerk beim Aufbau der mobilen Verbindung vergeben wurde (z.B. 100.x.x.x.x.x; die IP-Adresse ist für eine SIM-Karte immer gleich).

2 INSTALLATION VON OPENVPN AUF WINDOWS

Installieren Sie die OpenVPN Software von der OpenVPN Webseite:
https://openvpn.net/index.php/open-source/downloads.html

Laden Sie die Datei *client.ovpn* aus dem Abschnitt "Konfiguration" im Kundenportal herunter und speichern diese auf Ihrem Server unter „C:\Program Files\OpenVPN\config“.

Laden Sie die Datei *credentials.txt* aus dem Bereich "Konfiguration" im Kundenportal herunter und speichern diese in „C:\Program Files\OpenVPN\config".

Starten Sie die OpenVPN GUI-Anwendung und überwachen Sie die Verbindung. Für ein Logbuch über Ihre letzte Verbindung öffnen Sie die Logdatei in „C:\Program Files\OpenVPN\log\client.txt“.

(Alternativ auch in „C:\Users\<IHRPROFIL>\OpenVPN\log\client.txt“)

3 INSTALLATION VON OPENVPN AUF LINUX/MACOS

Installieren Sie OpenVPN mit `sudo apt-get install openvpn;` bei der Verwendung von „Ubuntu“ oder „Homebrew“ für MacOS mit `brew install openvpn`

Laden Sie die Datei „client.conf“ aus dem Kunden Portal herunter (Konfiguration -> OpenVPN Konfiguration -> Linux/macOS -> Download client.conf).

* Optional: Fügen Sie die Protokollzeile „/var/log/openvpn_1nce.log“ hinzu, nur für Debugging Zwecke.*

Kopieren Sie die Datei “client.conf” in /etc/openvpn.

Laden Sie die Datei „credentials.txt“ vom Kunden Portal herunter (Konfiguration -> OpenVPN Konfiguration -> Linux/macOS).

Kopieren Sie die Datei “credentials.txt” ebenfalls in /etc/openvpn.

Starten Sie die OpenVPN-Verbindung mit einer neuen Konfiguration:

```
sudo service openvpn restart
```

4 BEISPIEL EINES SETUP MIT ZWEI RASPBERRYS UND VERBINDUNGSTEST AUF LINUX

Die Testumgebung zeigt, wie man überprüft, ob die OpenVPN-Verbindung erfolgreich hergestellt wurde.

Die folgenden Testumgebungen wurden auf einem Linux-Server ausgeführt.

In dieser Testumgebung wurden zwei Raspberry Pi Geräte verwendet. Sie werden " raspberrypi2" und " raspberrypi3" genannt.

Die folgenden Schritte zeigen, wie Sie überprüfen können, ob die OpenVPN-Verbindung erfolgreich hergestellt wurde:

4.1 RASPBERRYPI2:

Das raspberrypi2 als datenempfangender Applikationsserver baut die OpenVPN-Verbindung auf.

Schauen Sie in die Logdatei, z.B. mit <sudo cat /var/log/openvpn_1nce.log>.

Nachfolgend die letzten Zeilen aus der Logdatei zu dem 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 diesem Beispiel ist 10.64.71.XX die Adresse des OpneVPN Client.

Geben Sie <ifconfig> ein und suchen Sie nach tun0. Im Test haben wir folgende Ergebnisse erzielt:

```
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:

Das raspberrypi3, in diesem Fall ein 3G USB Surfstick, Huawei E173, ist in dieser Testumgebung das Endgerät mit der 1NCE SIM Karte (Modem).

Mit “wvdial” gefolgt von “config” erhalten Sie eine Ansicht der Standardkonfiguration ihres Geräts:

```
$ cat /etc/wvdial.conf

[Dialer 1nce]
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2
Init3 = AT+CGDCONT=1,"IP","iot.1nce.net"
Modem Type = Analog Modem
Baud = 9600
New PPPD = yes
Modem = /dev/ttyUSB1
ISDN = 0
Phone = *99#
Password = *
Username = *
```

Bitte beachten Sie, dass diese Konfiguration auf dieser Testumgebung basiert. Wenn Sie ein anderes Modem nutzen, müssen Sie zuerst folgenden Befehl ausführen:

```
sudo wvdialconf
```

Anschließend setzen Sie den 1NCE APN und speichern die Konfiguration mit einem Namen Ihrer Wahl. In dieser Testumgebung wurde die config-Datei mit “1nce” benannt.

Eine Verbindung wird aufgebaut mit dem Befehl:

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

Wenn Sie in die Logdatei schauen, sehen Sie, dass die Verbindung hergestellt ist.

```
$ 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.
--> Starting pppd at Mon Aug 27 15:35:54 2018
--> Pid of pppd: 12919
--> Using interface ppp0
--> pppd: [1a]
--> pppd: [1a]
--> pppd: [1a]
--> pppd: [1a]
--> pppd: [1a]
--> 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]
```

Die private IP-Adresse ist in diesem Fall 100.117.XXX.5, welche die IP-Adresse des Gerätes ist.

Jetzt haben Sie eine direkte Verbindung durch den OpenVPN-Tunnel zum raspberrypi2, dem Anwendungsserver in dieser Testumgebung.

Mit einem ping-Befehl können Sie die Verbindung vom Server (raspberrypi2) zum Endgerät (raspberrypi3) überprüfen:

```
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
```

Die Verbindung vom Endgerät (raspberrypi3) zum Server (raspberrypi2) kann auch mit einem ping-Befehl überprüft werden:

```
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
```

Nachdem Sie die OpenVPN-Verbindung hergestellt und getestet haben, können Sie die Verbindung auf dem raspberrypi2 mit folgendem Befehl beenden:

```
$ sudo killall wvdial
$ tail -f /tmp/3G_1nce.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
```

5 FEHLERBEHEBUNG

5.1 LOKALISIERUNG DER LOGDATEIEN

Speicherort der Protokolldatei für den OpenVPN Connect Client für Windows:

```
C:\Program Files (x86)\OpenVPN Technologies\OpenVPN
Client\etc\log\openvpn_(unique_name).log
```

Standardmäßig geht die OpenVPN-Protokollausgabe in den meisten Linux-Distributionen an das syslog, das normalerweise unter folgendem Pfad liegt:

```
/var/log/syslog
```

Speicherort der Protokolldatei für den OpenVPN Connect Client für MacOS:

```
/Library/Application Support/OpenVPN/log/openvpn_(unique_name).log
```

Macintosh zeigt Ihnen den entsprechenden Ordner möglicherweise nicht mit der Suchfunktion an, da diese Ihnen nur bestimmte Dinge anzeigt und andere verbirgt. Um also zum Ordner „/Library“ zu gelangen, öffnen Sie die Suchfunktion und wählen Sie im Menü oben „Go“, gefolgt von „Go to folder“ und geben Sie dann den Pfad „/Library“ ein, um in dieses Verzeichnis zu gelangen. Sie können dann in den richtigen Ordner wechseln und die Protokolldatei einsehen. Bitte beachten Sie auch, dass der OpenVPN Connect Client für Macintosh nur über bestimmte Berechtigungen in der Protokolldatei verfügt, so dass Sie ihn nicht normal öffnen können. Um dies zu umgehen, klicken Sie mit der rechten Maustaste auf die Protokolldatei und wählen Sie im Menü die Option "Get info". Dann können Sie unten, unter "Sharing & Permissions", auf das gelbe Vorhängeschloss-Symbol klicken, um die Einstellungen freizuschalten und allen Lesezugriff zu gewähren. Dann können Sie die

Protokolldatei mit einem Rechtsklick öffnen, indem Sie "Öffnen mit" wählen und dann einen Texteditor oder Ähnliches wählen, um den Inhalt der Protokolldatei anzuzeigen.

5.2 ROUTINGTABELLE FÜR OPENVPN

Jeder Kunde erhält bei der ersten Bestellung ein volles /22 IP-Subnetz zugewiesen. Abhängig von der Anzahl der SIM-Karten werden weitere Subnetze hinzugefügt.

Der OpenVPN-Client zieht während der Initialisierung der Verbindung die Informationen für alle erforderlichen Routes auf den Systemen. Es wird daher empfohlen, den OpenVPN-Client regelmäßig neu zu starten, um sicherzustellen, dass alle neuen Subnetze in die lokalen Routingtabellen aufgenommen werden.

5.3 POTENZIELLER ROUTE-SUBNETZ-KONFLIKT

Der OpenVPN-Protokolleintrag "Potential Route Subnet Conflict" bedeutet, dass Ihre SIM-Karten IP-Adressen haben, die auch in Ihrem lokalen Netzwerk vorhanden sind.

Eine Möglichkeit, dies zu beheben besteht darin, eine Option "redirect gateway local" in die OpenVPN-Konfigurationsdatei aufzunehmen. Eine weitere Möglichkeit, dies zu beheben besteht darin, die Adressen Ihres lokalen LANs zu ändern. Dazu ändern Sie die Konfiguration Ihres Routers. Bei einigen Routern geben Sie die ersten drei Zahlen des LAN an (z.B. 192.168.77); bei anderen Routern geben Sie die Adresse des Routers selbst an (z.B. 192.168.77.1).

5.4 OPENVPN VERBINDUNG OHNE GRUND ABGEBROCHEN

Der OpenVPN-Client unterstützt nur eine aktive Verbindung zum OpenVPN-Server. Oft werden die vorgegebenen Anmeldeinformationen bereits für eine andere Verbindung verwendet werden. Dies wird in der Regel sichtbar, wenn die Protokolle eine kontinuierliche Wiederverbindung und ein Abbrechen der Verbindung zeigen.

Bitte prüfen Sie, ob ein anderer Benutzer auch versucht, eine OpenVPN-Verbindung mit den angegebenen Zugangsdaten herzustellen.

5.5 HTTPS TIMEOUT DURCH OPENVPN VERBINDUNG

HTTPs-Verbindungen zu Ihrem Anwendungsserver können bei einer Verbindung über OpenVPN ausfallen.

Die Änderung der MTU-Größe für die Netzwerkschnittstelle kann diese Herausforderung lösen. Aufgrund der begrenzten Bandbreite von 128 Kbit/s verursacht die MTU-Standardgröße (1500) diesen Fehler. Die Änderung der MTU-Größe auf etwas weniger als 1500, wie beispielsweise 1400, sollte dies beheben.

5.6 OPENVPN AUTHENTIFIZIERUNGSFEHLER

Bestimmte OpenVPN-Versionen unterstützen keine Passwörter länger als 128 Zeichen. Daher schlägt in diesem Fall die Authentifizierung fehl.

Dieser Fehler ist z.B. im "OpenVPN 2.4.6 arm-openwrt-linux-gnu" auf OpenWRT mit deaktivierter PKCS#11 bekannt. Die Neukompilierung von OpenVPN nach dem Patchen der Unterstützung für längere Passwörter behebt diesen Fehler. Der Fehler ist in der OpenVPN-Community bekannt und kann unter folgendem Link noch einmal ausführlich nachgelesen werden: https://community.openvpn.net/openvpn/ticket/712