OpenVPN Integration Guide for Linux

wherever SIM customers can create their own Virtual Private Network for their mobile IoT/M2M devices fitted with wherever SIMs. Data traffic will be exchanged between the devices and the application server through an OpenVPN tunnel, enabling direct communication with the IPs of the mobile devices (no NAT applied).

The tunnel is established between the wherever SIM Core Network and the customer's VPN gateway or server.

Any traffic exchanged with the mobile devices is encrypted before transmitted over the public internet, therefore adding an additional layer of security and privacy. For that, no VPN software needs to be installed on the device.

Setting Up OpenVPN Client on Linux/Ubuntu

Change Internet Breakout Region in "Service Profile" in portal

Log in with your user account to the wherever SIM Portal and select "Service Profiles". Mark the Service Profile you will choose for devices with VPN and click on the tab "Basic Configuration". Change the Internet Breakout Region to "eu-west-1 (VPN)" or your preferred region (VPN).

Install OpenVPN Software

Install openvpn package

```
sudo apt-get install openvpn
```

Download and Install VPN Configuration File

Log in with your user account on the wherever SIM Portal and select the "Link" Icon on the top/right. From there you can download a pre-configured configuration file, the filename is **client.conf**

Please store that file on your server in the folder /etc/openvpn.

Create Credentials for Authentication

In the next steps you need to create a file called credentials.txt in the folder /etc/openvpn. You can choose to use your user credentials to authenticate or to use an application token (recommended).

Authentication with User Credentials

The content of the credentials.txt must be just two lines, first line your username and second your password.

```
username@domain.com
YourPassword
```

Authentication with Application Token

If you do not want to store your credentials you can also choose to enter them each time the VPN tunnel is established, if you prefer that option please comment out the line "auth-user-pass /etc/openvpn/credentials.txt" in the client.conf file.

When you run the OpenVPN client on a VPN gateway or application server it is recommended to use a dedicated application token. In this case the first line in the credentials.txt file needs to be filled with your wherever SIM's organisation identifier and instead of the password you store the application token.

You can create application tokens when you log in to the wherever SIM Portal, select the "Link" Icon on the top/right and then "Create New Application Token". Please Copy+Paste the token into the credentials file.

The content of the credentials file would then look like this

```
orgId
ApplicationToken
```

When you log in to the wherever SIM Portal it will shows this data under the VPN settings.

Protecting the Credentials File

You should keep the credentials.txt file only readable by root and not by other users of your server. You can ensure this with the following commands:

```
sudo chown root.root /etc/openvpn/credentials.txt
sudo chmod 600 /etc/openvpn/credentials.txt
```

Starting and Monitoring the OpenVPN connection

Now you can start the VPN client by running

```
sudo service openvpn start
```

The openvpn daemon will log into /var/log/syslog, if everything works it would like this:

```
Jul 12 17:53:55 openvpn-client ovpn-client[3027]: Data Channel Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Jul 12 17:53:55 openvpn-client ovpn-client[3027]: Data Channel Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Jul 12 17:53:55 openvpn-client ovpn-client[3027]: Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 2048 bit RSA
Jul 12 17:53:55 openvpn-client ovpn-client[3027]: [openvpn.whereversim.com] Peer Connection Initiated with [AF_INET]52.209.x.y:1194
Jul 12 17:53:57 openvpn-client ovpn-client[3027]: SENT CONTROL [openvpn.whereversim.com]: 'PUSH_REQUEST' (status=1)
Jul 12 17:53:57 openvpn-client ovpn-client[3027]: PUSH: Received control message: 'PUSH_REPLY,route 10.64.0.1,topology net30,ping 1,ping-restart 5,route 10.x.y.z 255.255.128.0,ifconfig 10.64.0.224 10.64.0.225'
Jul 12 17:53:57 openvpn-client ovpn-client[3027]: TUN/TAP device tun0 opened
Jul 12 17:53:57 openvpn-client ovpn-client[3027]: /sbin/route add -net 10.64.0.1 netmask 255.255.255.255 gw 10.64.0.225
Jul 12 17:53:57 openvpn-client ovpn-client[3027]: /sbin/route add -net 10.x.y.z netmask 255.255.128.0 gw 10.64.0.225
```

Finding the static private IP of your VPN client

The wherever SIM OpenVPN server will allocate a static IP address to the tun interface of your VPN client, this IP will also stay the same when your VPN client is reconnecting or if you move the tunnel to a different machine. So you can use it on your mobile devices to address your application, nevertheless you should never configure the IP directly on your devices, but use a DNS to resolve it.

Once the tunnel is established, you can see that IP address on your tun interface:

```
ubuntu@openvpn-client:~$ ip a s tun0
14: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 100
    link/none
    inet 10.64.0.224 peer 10.64.0.225/32 scope global tun0
```

In this sample the IP address is 10.64.0.224

Testing Successful Data Connectivity

If the VPN tunnel is established successfully you will be able to connect directly to the private IP addresses of your mobile devices.

For testing you can choose any for your endpoints that as currently an active data session. for log in to the wherever SIM Portal and select on of your endpoints, in the details you will if it is currently online or not and you will see the IP address it has assigned.

Now will be able to ping that private IP address:

```
root@openvpn-client:~# ping 10.x.y.z
PING 10.x.y.z (10.x.y.z) 56(84) bytes of data.
64 bytes from 10.x.y.z: icmp_req=1 ttl=62 time=72 ms
64 bytes from 10.x.y.z: icmp_req=2 ttl=62 time=80 ms
64 bytes from 10.x.y.z: icmp_req=3 ttl=62 time=75 ms
```

For this to work your device needs to run an IP stack that is responding to ICMP echo request, this might not be the case for embedded devices that do implement only partial IP stack functionality.

You will be able to use any network protocols, e.g. if your device is running a sshd daemon you would now be able to log into it via ssh.

Enabling Access for Servers behind the VPN client

If you have multiple servers behind your VPN gateway that need to communicate with your mobile device, you can apply masquerading using iptables to hide them behind the single IP address of your VPN client.

First you need to enable IP forwarding on your VPN gateway (if not already active) by editing your /etc/sysctl.conf and set net.ipv4.ip_forward=1, after that load the config with

```
sudo sysctl -p /etc/sysctl.conf
```

After that you need to enable masquerading for the tun interface by running

```
sudo iptables -t nat -A POSTROUTING -o tun0 -j MASQUERADE
```

You can also forward connections coming to specific ports to an application server behind your VPN gateway by using DNAT, e.g. to forward to an internal HTTP server at IP 192.168.100.21 use:

```
sudo iptables -t nat -A PREROUTING -i tun0 -p tcp --dport 80 -j DNAT --to-destination 192.168.100.21
```

If you want to make these settings persistent you can use the iptables-persistent package.