Grandstream Networks, Inc.

OpenVPN on Grandstream IP Phones Guide

Table of Contents

Table of Figures

SUPPORTED DEVICES

Following table shows Grandstream IP Phones supporting OpenVPN feature:

Model	Supported	Firmware
Small Business IP Phones GXP16XX Series		
GXP1610/15	Yes	1.0.4.6 or higher
GXP1620/25		
GXP1628		
GXP1630		
Mid-Range IP Phones GXP17XX Series		
GXP1760	Yes	1.0.0.37 or higher
GXP1780/82		
Enterprise IP Phones GXP21XX Series		
GXP2130	Yes	1.0.7.25 or higher
GXP2140		
GXP2160		
GXP2135		
GXP2170		

INTRODUCTION

A Virtual Private Network (VPN) is used to create an encrypted connection enabling users to send and receive data across shared or public networks acting as clients connected to a private network. The benefit of using a VPN is to ensure the appropriate level of security to connected systems when the underlying network infrastructure alone cannot provide it. The most common types of VPNs are remote-access VPNs and site-to-site VPNs.

VPNs can be defined between specific end points such as IP-Phones and computers, and also servers in separate data centers, when security requirements for their exchanges exceed what the enterprise network can deliver. Increasingly, enterprises use VPNs to secure data and voice exchange.

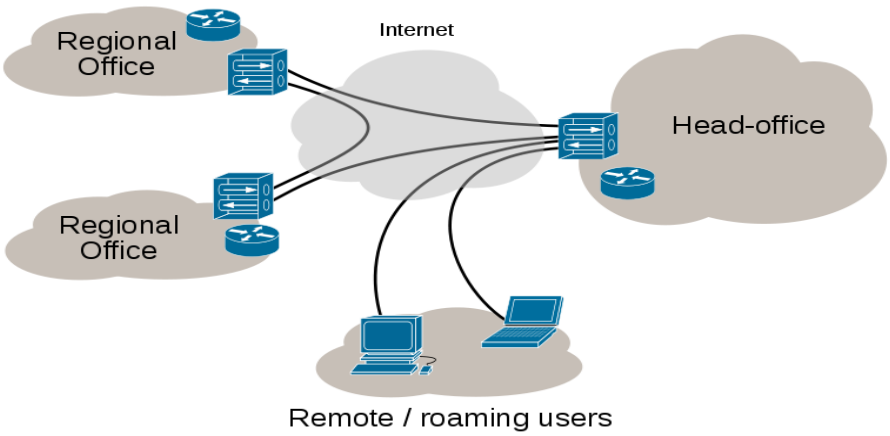

Figure 1: VPN Architecture Overview

The VPN security model provides:

- Client authentication to forbid any unauthorized user from accessing the VPN network.
- Encryption and confidentiality that will prevent man in middle attacks and eavesdropping on the network traffic.
- Data integrity to maintain the consistency, and trustworthiness of the messages exchanged.

Users must be authenticated before establishing secure VPN tunnels. Client/server tunnels use passwords or digital certificates. It is possible to permanently store the key to allow the tunnel to be established automatically.

The purpose of this guide is to underline OpenVPN client feature on Grandstream IP Phones with an OpenVPN server.

ABOUT OPENVPN

OpenVPN is an open-source software application that implements virtual private network (VPN) techniques for creating secure point-to-point or site-to-site connections in routed or bridged configurations and remote access facilities. It uses a custom security protocol that utilizes SSL/TLS for key exchange. It is capable of traversing network address translators (NATs) and firewalls.

OpenVPN offers:

- Tunneling any IP subnetwork or virtual Ethernet adapter over a single UDP or TCP port.
- Using all of the encryption, authentication, and certification features of the OpenSSL library to protect your private network traffic as it transits the internet.
- Using any cipher, key size, or HMAC digest (for datagram integrity checking) supported by the OpenSSL library.
- Choosing between static-key based conventional encryption or certificate-based public key encryption.
- Using static, pre-shared keys or TLS-based dynamic key exchange.
- Tunneling networks whose public endpoints are dynamic such as DHCP or dial-in clients.
- Tunneling networks over NAT and through connection-oriented stateful firewalls without having to use explicit firewall rules.

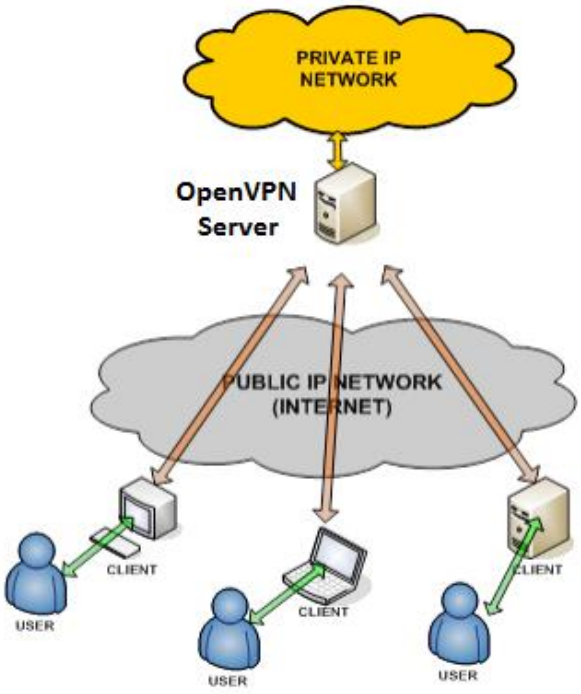

Figure 2: OpenVPN Architecture

OPENVPN ON GRANDSTREAM PHONES

OpenVPN feature is supported on Grandstream Small and Enterprise IP Phones Series (GXP16xx and GXP21xx Series), to give the ability to connect as clients to the OpenVPN server through VPN tunnel over a shared or public network to secure both voice and data exchange, remote clients will be considered as in the same private network allowing point to point calls, access the phone's configuration…

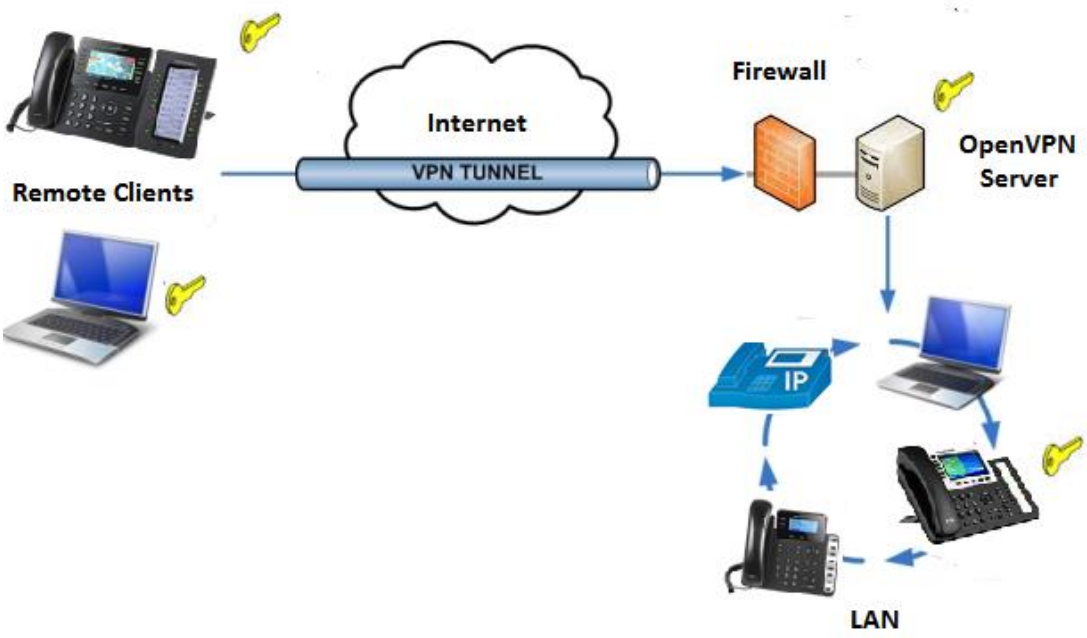

Figure 3: Grandstream Phones with OpenVPN

OPENVPN SERVER INSTALLATION AND CONFIGURATION

OpenVPN Server is available for Linux, Windows operating systems and routers supporting OpenVPN.

This guide will cover installation and configuration steps including Certificates generation supported by Grandstream IP Phones using OpenVPN on Linux Ubuntu 12.04 distribution.

Server Installation

1. To install OpenVPN server, open a new terminal and login as root.
2. Enter "**apt-get install openvpn**" to download and install the server as shown in screenshot below.
3. A prompt will pop up to confirm the action, confirm by typing **Y**.

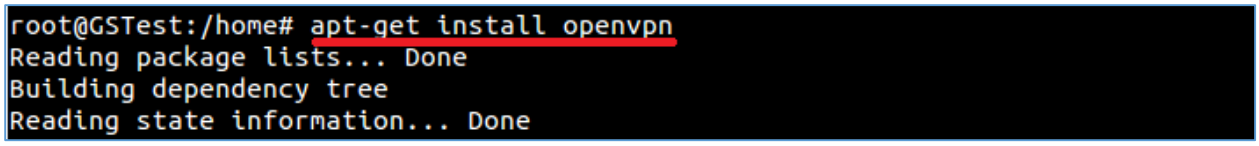

Figure 4: Installing OpenVPN

Server Configuration

OpenVPN server configuration is based on **server.conf** file that needs to be created.

Following instructions and table show needed steps to create configuration file and explaining options to use.

1. Type **nano /etc/openvon/server.conf** in order to create empty new configuration file.
2. Type in configuration options as shown in following screenshot. Configuration options are explained in next table below.

```
 GNU nano 2.2.6                File: server.conf                      Modified

port 1194
proto udp
dev tun
ca ca.crt
cert server.crt
key server.key
dh dh1024.pem
server 10.10.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "redirect-gateway def1"
client-to-client
cipher BF-CBC
comp-lzo
duplicate-cn
keepalive 10 120
max-clients 100
status openvpn-status.log
log         openvpn.log
verb 7
```

Figure 5: Server.conf

3. Save and exit using "**Ctrl+X**" then confirm with "**Y**".

The table below gives an overview of OpenVPN server configuration options.

Option	Use / example
port	TCP/UDP port that OpenVPN should listen on, please make sure to open up this port on the firewall *Example:* **port 1194**
proto	TCP or UDP that will be used on the server. *Example:* **proto udp**
dev	TUN/TAP device: TUN device is a virtual IP point-to-point device. TAP device is a virtual Ethernet device when bridging Ethernet interface**.** *Example:* **dev tun**
ca	SSL/TLS certificate authority (ca) that will be used by the server. *Example:* **ca.crt**
cert	Server certificate (cert) that will be used by the server. *Example:* **server.crt**
key	Private key (key) that will be used by the server *Example:* **server.key**
dh	Private handshake key between the server and the client 1024 or 2048 bit keys. *Example:* **dh1024.pem**
server	VPN subnet for OpenVPN to draw client addresses from. *Example:* **10.10.0.0 255.255.255.0**
ifconfig-pool-persist ipp.txt	Maintain a record of client <-> virtual IP address associations in this file. *Example:* **ifconfig-pool-persist ipp.txt**

Push	Forces all clients to redirect their default network gateway through the VPN, causing all IP traffic such as web browsing and DNS lookups to go through the VPN. *Example:* **push "redirect-gateway def1"**
client-to-client	Clients to be able to "see" each other, useful for point to point calling. *Example:* **client-to-client**
duplicate-cn	Same generated certificates and key can be used by all clients. *Example:* **duplicate-cn**
keepalive	Ping every 10 seconds to check remote peer status. If no ping received after 120 seconds, the remote peer will be considered as down. *Example:* ***keepalive* 10 120**
cipher	Cryptographic cipher BF-CBC, AES-128-CBC or DES-EDE3-CBC. *Example:* **cipher BF-CBC**
comp-lzo	Enable compression on the VPN link. *Example:* **comp-lzo**
max-clients	The maximum number allowed of connected clients. *Example:* **max-clients 100**
status openvpn-status.log	Output a short status file showing current connections, rewritten every minute. *Example:* **status openvpn-status.log**
log openvpn.log	Log messages, "log" will truncate the log file on OpenVPN startup, while "log-append" will append to it. *Example:* **log openvpn.log**
verb	Level of log file verbosity: 0 is silent, except for fatal errors 4 is reasonable for general usage 5 and 6 can help to debug connection problems 9 is extremely verbose *Example:* **verb 7**

Generating Server/Clients Certificates

After installing and configuring the server, we need to generate server and clients' certificates.

Prerequisites

Easy-rsa tool is needed in order to generate server and client certificates.

1. Create **easy-rsa** directory under **/etc/openvpn** directory.

```
root@GSTest:/home# mkdir /etc/openvpn/easy-rsa
```

Figure 6: Create easy-rsa Directory

2. Copy the content of easy-rsa tool on the OpenVPN directory as shown below:

```
cp -r /usr/share/doc/openvpn/examples/easy-rsa/* /etc/openvpn/easy-rsa/
```

Figure 7: Copy easy-rsa Content

Once the **easy-rsa** directory copied, two-sub directories **1.0** and **2.0** under "**/etc/openvpn/easy-rsa**" will be found.

3. Access **2.0** directory by typing **cd /etc/openvpn/easy-rsa/2.0** and follow below instructions to generate the certificates.

Creating Server Certificates

2.0 directory contains necessary files to generate the server and client certificates, the file **vars** contains information about the certificate, such as the Key size, the expiration delay of the certificate and other settings (country, city, mail…) as shown below.

```
export KEY_SIZE=1024

# In how many days should the root CA key expire?
export CA_EXPIRE=3650

# In how many days should certificates expire?
export KEY_EXPIRE=3650

# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Fort-Funston"
export KEY_EMAIL="me@myhost.mydomain"
export KEY_EMAIL=mail@host.domain
export KEY_CN=changeme
export KEY_NAME=changeme
export KEY_OU=changeme
export PKCS11_MODULE_PATH=changeme
export PKCS11_PIN=1234
```

Figure 8: Vars File

1. Modify the file by entering appropriate parameters using command: **nano vars.** Then save and exit using "**Ctrl+X**" and confirm the changes with **Y**.
2. Type the following command: **source ./vars** to apply this file settings for server and client certificates configuration.

```
root@GSTest:/etc/openvpn/easy-rsa/2.0# source ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/
2.0/keys
```

Figure 9: Apply Config to the Server

3. Type **"./Clean-all"** command as requested.

```
./clean-all
```

Figure 10: Clean-all

4. Type "**./build-dh**" command to build deffie-hellman parameters required by the server for SSL/TLS connection.

Figure 11: Build dh

5. Generate the Certification Authority (CA) and the Server Key (server.key) using following commands "**./pkitool --initca**" and "**./pkitool --server server**" respectively as shown in below screenshot:

```
root@GSTest:/etc/openvpn/easy-rsa/2.0# ./pkitool --initca
Using CA Common Name: server
Generating a 1024 bit RSA private key
.....................++++++
.++++++
writing new private key to 'ca.key'
-----
root@GSTest:/etc/openvpn/easy-rsa/2.0# ./pkitool --server server
Using Common Name: server
Generating a 1024 bit RSA private key
.........++++++
..............................++++++
writing new private key to 'server.key'
-----
Using configuration from /etc/openvpn/easy-rsa/2.0/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'US'
stateOrProvinceName   :PRINTABLE:'CA'
localityName          :PRINTABLE:'SanFrancisco'
organizationName      :PRINTABLE:'Fort-Funston'
organizationalUnitName:PRINTABLE:'server'
commonName            :PRINTABLE:'server'
name                  :PRINTABLE:'server'
emailAddress          :IA5STRING:'mail@host.domain'
Certificate is to be certified until Sep  5 08:55:45 2026 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated
```

Figure 12: Generating Keys

6. Copy generated certificates to "**/etc/openvpn**" directory (generated keys can be found under "**/etc/openvpn/easy-rsa/2.0/keys**" as shown below:

```
root@GSTest:/home# cd /etc/openvpn/easy-rsa/2.0/keys/
root@GSTest:/etc/openvpn/easy-rsa/2.0/keys#
root@GSTest:/etc/openvpn/easy-rsa/2.0/keys# cp dh1024.pem server.crt server.key ca.crt /etc/openvpn/
```

Figure 13: Copy Certificates

7. If more client keys are created, every client needs a unique common name. For extra security, generate a key to use with `tls-auth` which adds an additional HMAC signature to all SSL/TLS handshake packets. Generate this key using following command:

```
openvpn --genkey --secret ta.key
```

Figure 14: Generating ta.key

Generating Client Certificates

In order to establish connection with OpenVPN server, clients need to use Clients certificates.
Following steps show how to generate them:

1. On a terminal enter "**/etc/openvpn/easy-rsa/2.0**" directory by typing "**cd /etc/openvpn/easy-rsa/2.0**".

2. Modify **vars** file by entering appropriate parameters (it needs to be different than server's vars file) using command: **nano vars.** Then save and exit using "**Ctrl+X**" and confirm the changes with **Y**.

3. Apply the configuration on the **vars** file to generate the client certificates using command "**source ./vars"**

```
root@GSTest:/etc/openvpn/easy-rsa/2.0# source ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/
2.0/keys
```

Figure 15: Apply Config to the Client Certificates

4. Generate client certificate and key using **pkitool,** and enter the name of the client certificates as below:

```
root@GSTest:/etc/openvpn/easy-rsa/2.0# ./pkitool client
Using Common Name: client
Generating a 1024 bit RSA private key
.....++++++
......++++++
writing new private key to 'client.key'
-----
Using configuration from /etc/openvpn/easy-rsa/2.0/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'US'
stateOrProvinceName   :PRINTABLE:'CA'
localityName          :PRINTABLE:'SanFrancisco'
organizationName      :PRINTABLE:'Fort-Funston'
organizationalUnitName:PRINTABLE:'client'
commonName            :PRINTABLE:'client'
name                  :PRINTABLE:'client'
emailAddress          :IA5STRING:'mail@host.domain'
Certificate is to be certified until Sep  5 08:59:18 2026 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated
root@GSTest:/etc/openvpn/easy-rsa/2.0#
```

Figure 16: Generating Client Certificates

The client certificates (**client.crt**) and key (**client.key**) will be generated and can be found under "**/etc/openvpn/easy-rsa/2.0/keys**"

CONFIGURING OPENVPN CLIENT ON GRANDSTREAM PHONES

On the phone side, users can use the phone's WebGUI to upload directly the certificates previously generated to connect to the OpenVPN server.
The following is applied to the all Grandstream Phones supporting OpenVPN.

Phone Settings

To activate OpenVPN feature on the phone, please follow below steps.

1. Log in to the WebGUI.
2. Go under "Maintenance > Network Settings".
3. Set "**OpenVPN Enable**" to Yes.
4. Enter the URL/FQDN of the OpenVPN Server on "**OpenVPN Server Address**", Port on "**OpenVPN Port**"
5. Choose TCP or UDP for "**OpenVPN Transport"** field according to OpenVPN server configuration file (UDP is used in this guide).
6. Click on Upload "**OpenVPN CA**" to upload "**ca.crt**".
7. Click on Upload "**OpenVPN Certificate**" to upload "**client.crt**".
8. Click on Upload "**OpenVPN Client key**" to upload "**client.key**".
9. Click on "Save" to save the changes.
10. Go under Account > Network Settings.
11. Set "**NAT Traversal"** to VPN.
12. Save and Apply the changes. Then reboot the phone.

The below screenshot shows an example of OpenVPN settings following this guide OpenVPN Server Configuration.

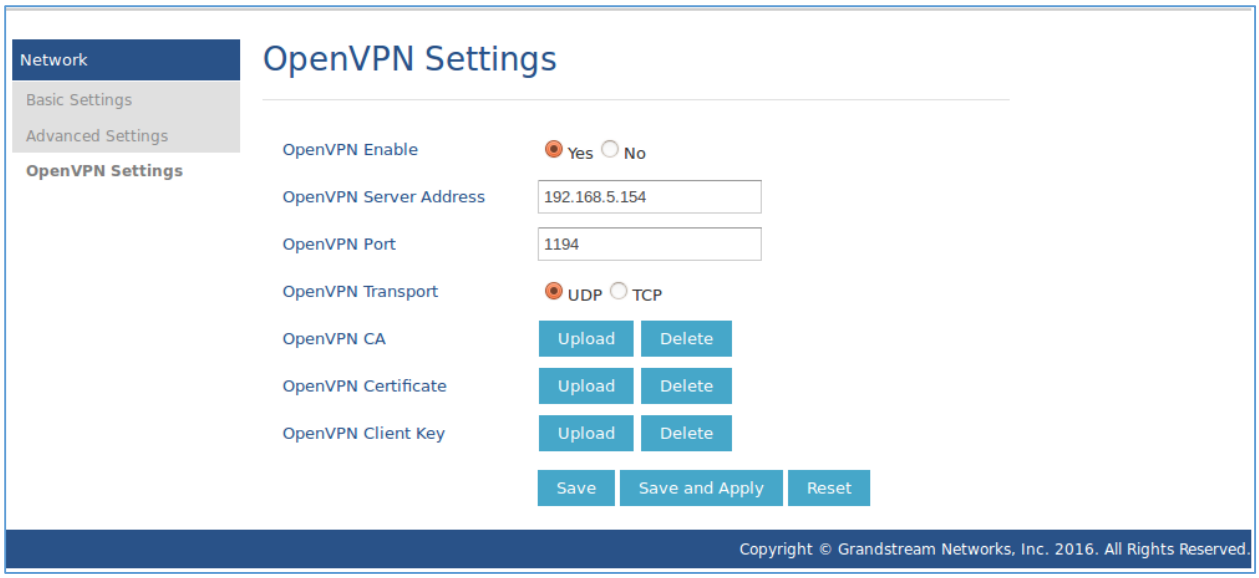

Figure 17: OpenVPN Client Settings

The phone will show two IP addresses for the shared private network, and for the OpenVPN network. Users can check network status from the phone's LCD menu or Web Interface.

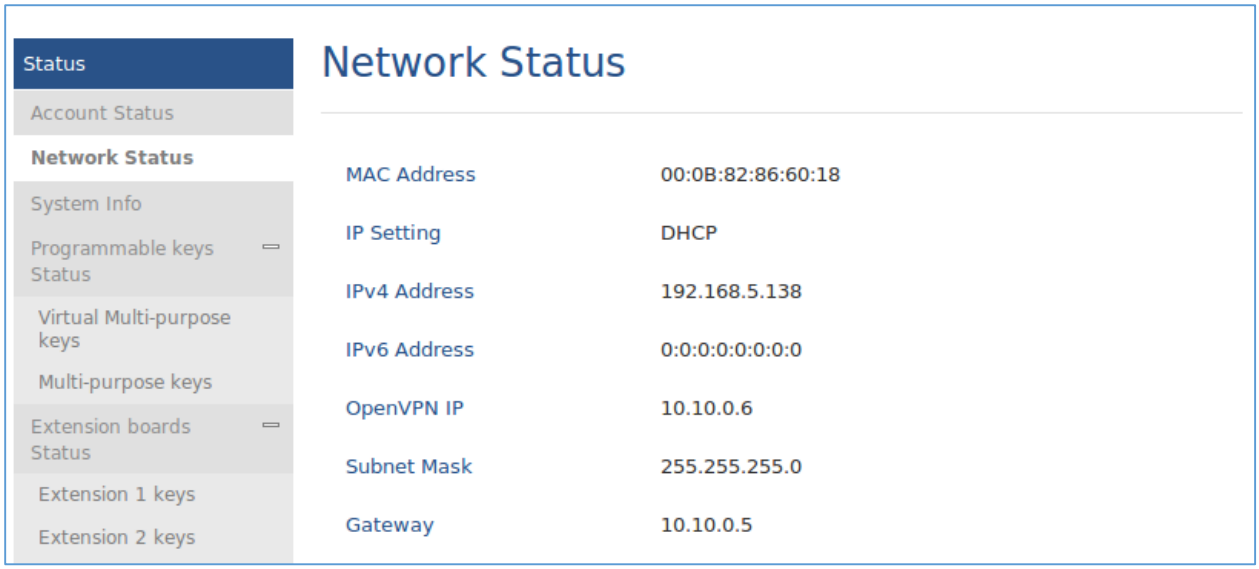

Figure 18: Network Status

Monitoring Clients

Users can check openvpn-status logs in order to monitor the client activities from the server side using this command "**tail –f /etc/openvpn/openvpn-status.log**" as shown in below screenshot:

```
root@GSTest:/etc/openvpn/easy-rsa/2.0/keys# tail -f /etc/openvpn/openvpn-status.log
OpenVPN CLIENT LIST
Updated,Wed Sep  7 10:27:05 2016
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
client,192.168.5.135:53620,11451,6228,Wed Sep  7 10:25:21 2016
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
10.10.0.6,client,192.168.5.135:53620,Wed Sep  7 10:27:02 2016
GLOBAL STATS
Max bcast/mcast queue length,0
END
```

Figure 19: OpenVPN-Status.log