Application Note 46

Configuring a TransPort WR router as an OpenVPN server for Windows OpenVPN clients

October 2020

Contents

1 INTRODUCTION

1.1 Outline

This document describes how to configure a Digi TransPort router as an OpenVPN server and how to configure Windows VPN clients.

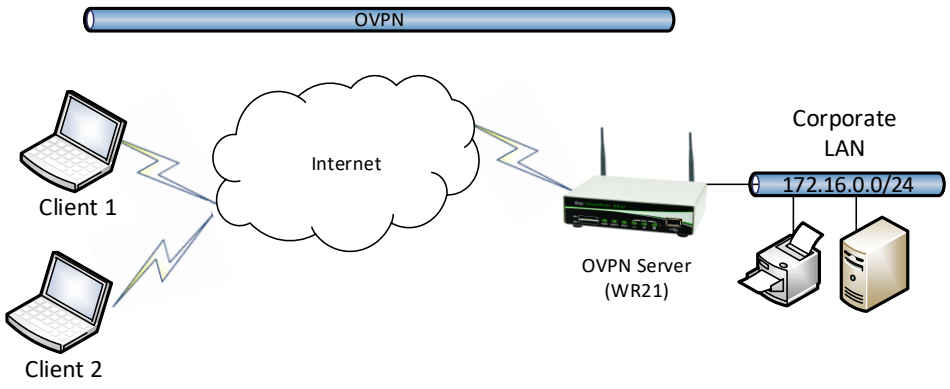
OpenVPN can be used for connecting to the router for secure management as well as access to services on the LAN side of the TransPort router, such as corporate messaging services, file servers and print servers for example.

From the OpenVPN website:

OpenVPN is a full-featured SSL VPN which implements OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or username/password credentials, and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface. OpenVPN is not a web application proxy and does not operate through a web browser.

OpenVPN 2.0 expands on the capabilities of OpenVPN 1.x by offering a scalable client/server mode, allowing multiple clients to connect to a single OpenVPN server process over a single TCP or UDP port.

For the purposes of this application note, the scenario consider 2 Remote clients (Windows laptop) connecting to the OVPN Server (TransPort WR)



1.2 Assumptions

This guide has been written for use by technically competent personnel with a good understanding of the communications technologies used in the product, and of the requirements for their specific application.

Configuration: This Application Note assumes the devices are set to their factory default configurations. Most configuration commands are only shown if they differ from the factory default.

This Application Note applies to:

Models shown: Digi TransPort WR21 router.

Software used: OpenVPN 2.2.2 and newer, Windows 10

Other Compatible Models: All other Digi TransPort WR products.

Firmware versions: 5130 or newer.

Acknowledgement: Much of the OpenVPN documentation has been taken directly from the HOWTO pages at the OpenVPN website.

Please see http://openvpn.net/index.php/open-source/documentation/howto.html for more details

1.3 Corrections

Requests for corrections or amendments to this Application Note are welcome and should be addressed to: tech.support@digi.com

Requests for new Application Notes can be sent to the same address.

1.4 Version

Version Number	Status
1.0	Published
1.1	Updated for new GUI
1.2	Updated screenshots for new web interface, rebranding (Oct 2016)
2.0	Updated for new version of OVPN, added static routes for Client LANs, added tests, adjust layouts and other fixes
2.1	October 2020 update: Added notes for OpenVPN client 2.4 and newer.

2 OPENVPN & EASY-RSA SETUP

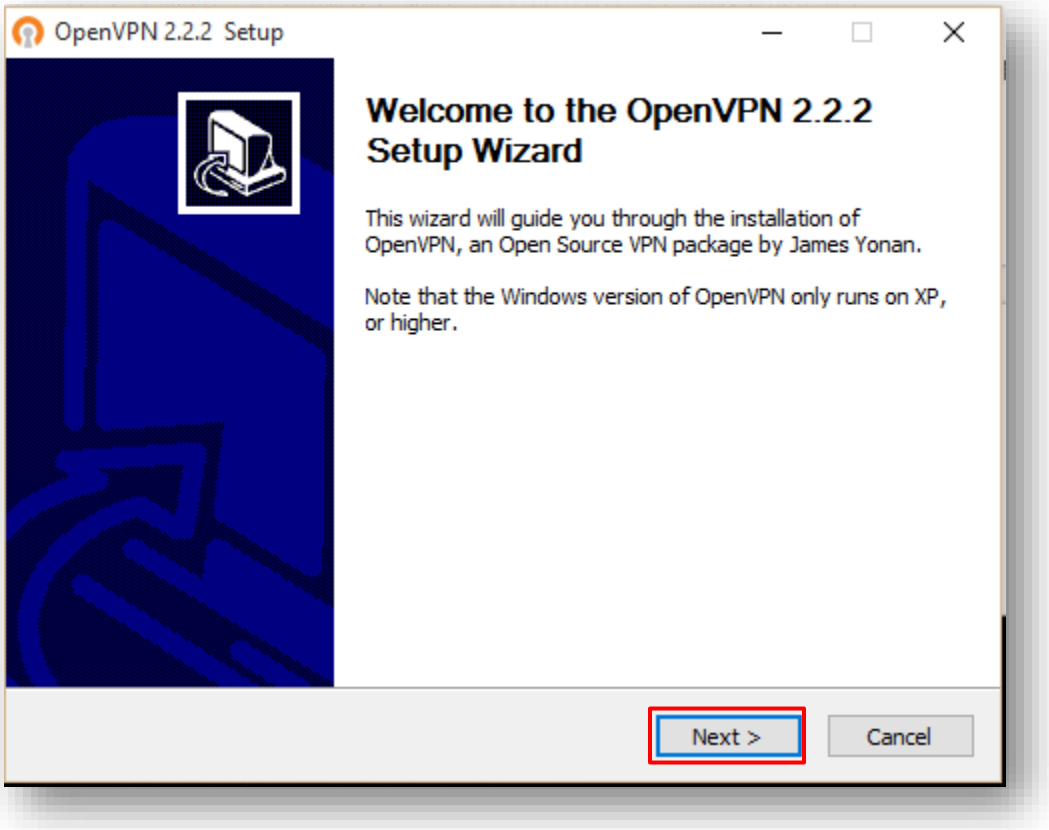
2.1 Download the OpenVPN installation package and install the software

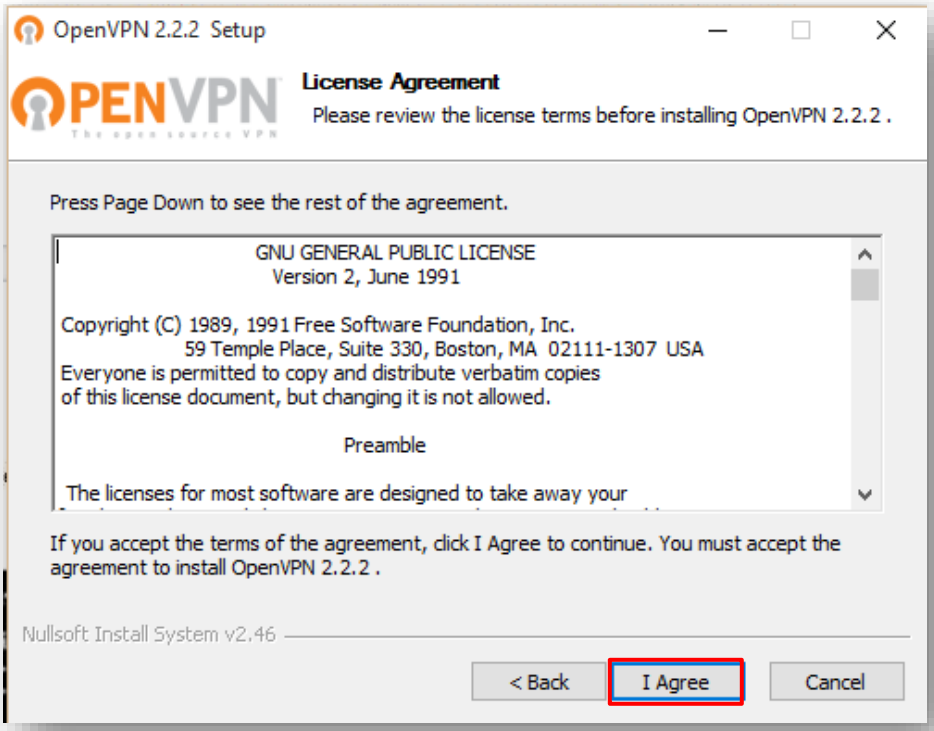
This step should be done on a PC that will be used to create the certificates.

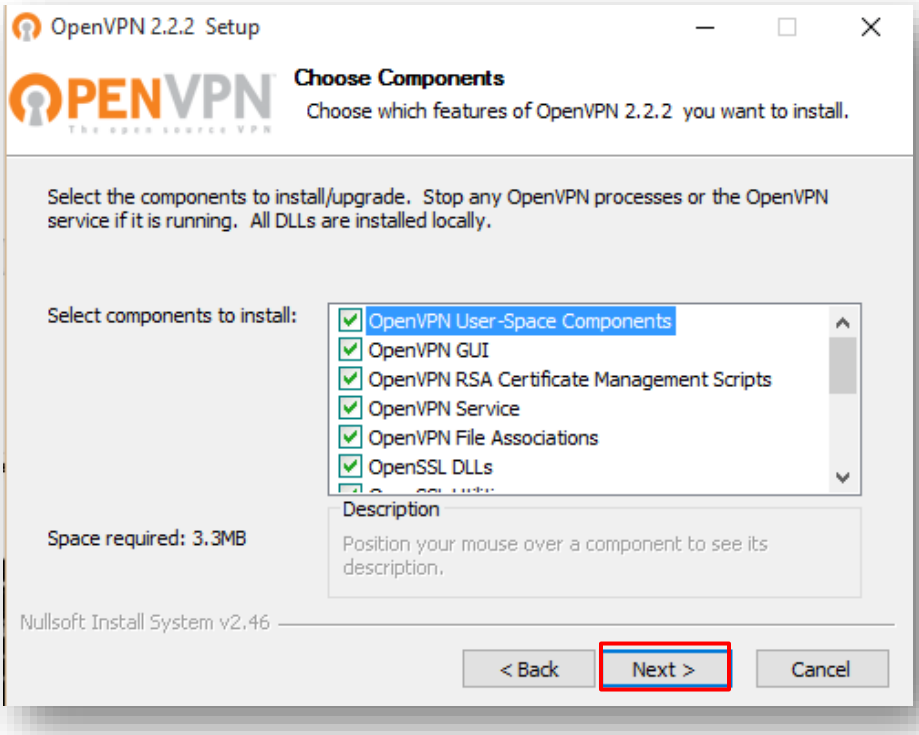
In order to download the installer, go to https://openvpn.net/community-downloads/ or here for older versions: https://build.openvpn.net/downloads/releases/ .

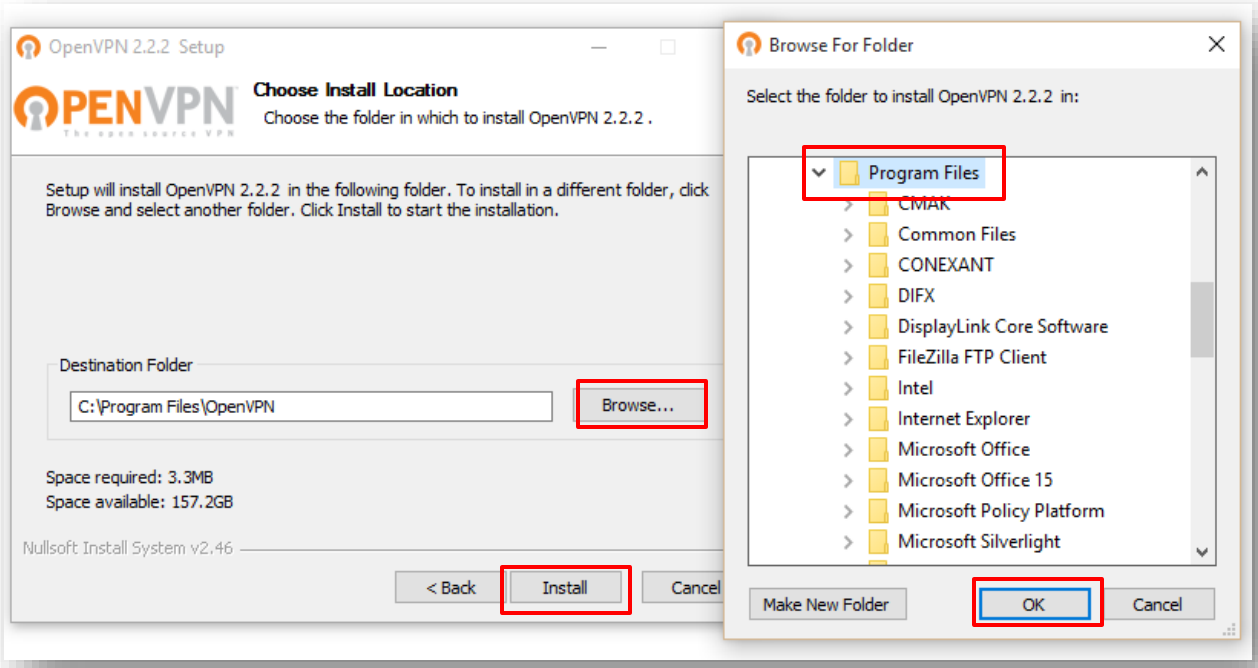
For this example, OpenVPN 2.2.2 version has been used.

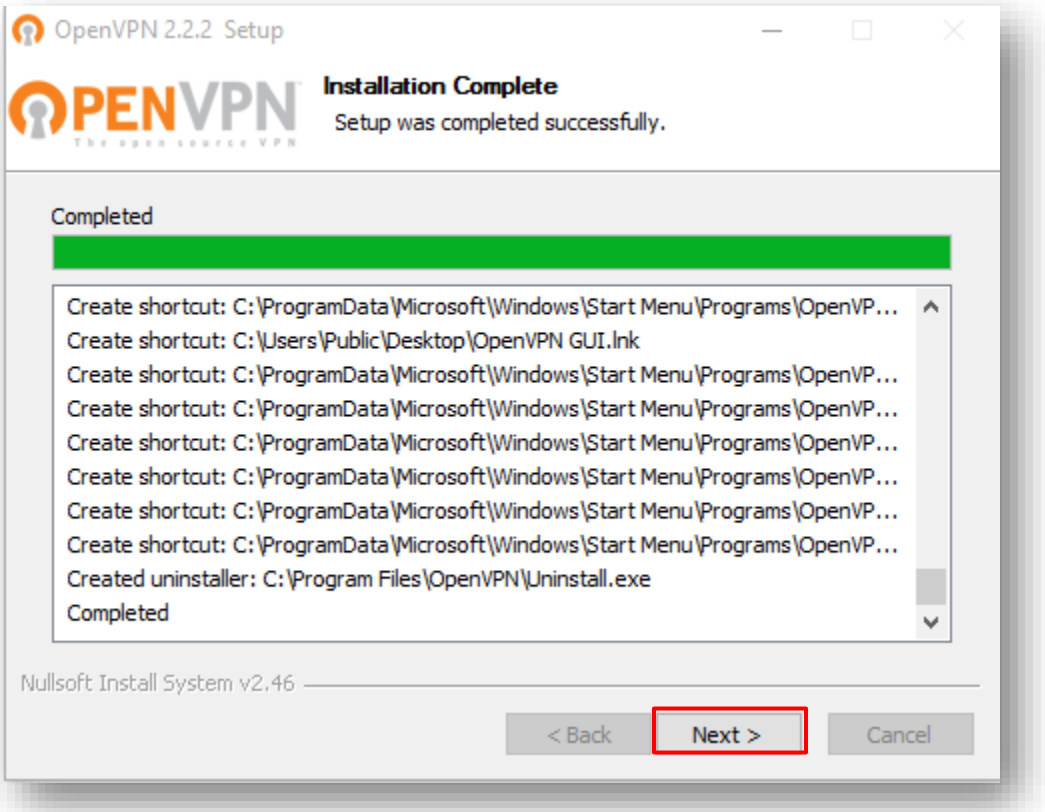
Run the installer and follow the instructions:

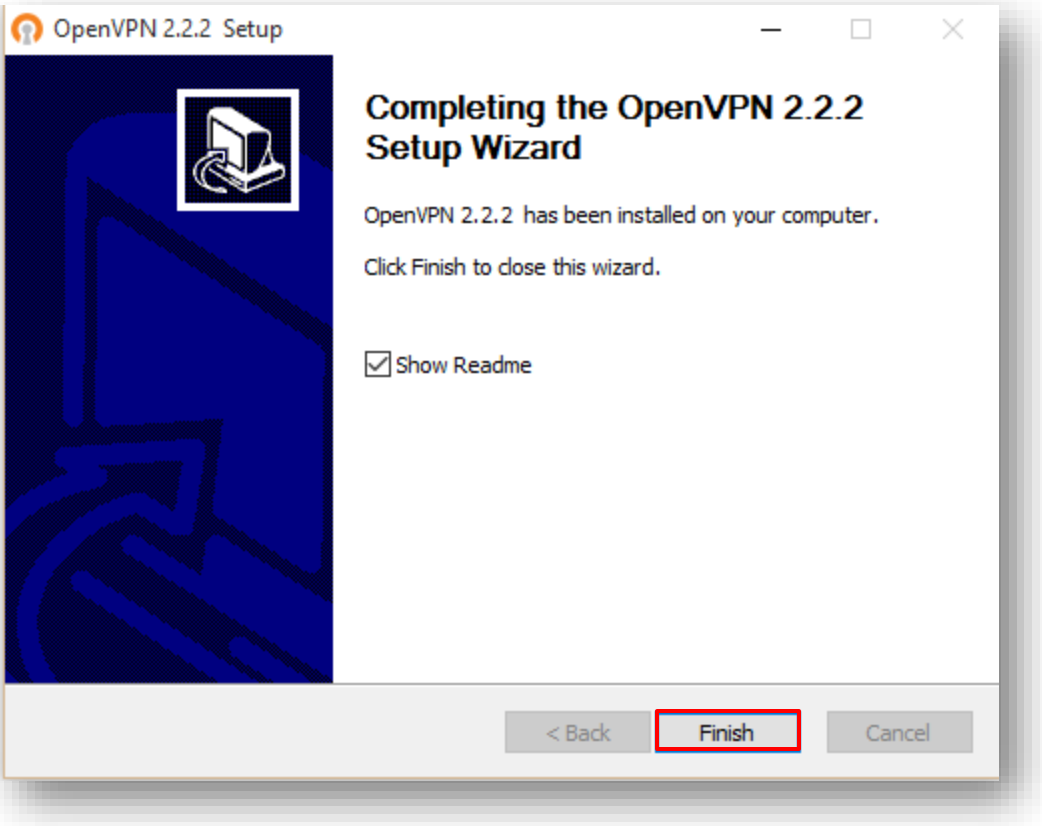



OpenVPN 2.2.2 Setup
License Agreement
Please review the license terms before installing OpenVPN 2.2.2 .
Press Page Down to see the rest of the agreement.
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
The licenses for most software are designed to take away your
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install OpenVPN 2.2.2 .
Nullsoft Install System v2.46
< Back
I Agree
Cancel


OpenVPN 2.2.2 Setup
Choose Components
Choose which features of OpenVPN 2.2.2 you want to install.
Select the components to install/upgrade. Stop any OpenVPN processes or the OpenVPN service if it is running. All DLLs are installed locally.
Select components to install:
OpenVPN User-Space Components
OpenVPN GUI
OpenVPN RSA Certificate Management Scripts
OpenVPN Service
OpenVPN File Associations
OpenSSL DLLs
Description
Position your mouse over a component to see its description.
Space required: 3.3MB
Nullsoft Install System v2.46
< Back
Next >
Cancel


OpenVPN 2.2.2 Setup
Choose Install Location
Choose the folder in which to install OpenVPN 2.2.2 .
Setup will install OpenVPN 2.2.2 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
C:\Program Files\OpenVPN
Browse...
Space required: 3.3MB
Space available: 157.2GB
Nullsoft Install System v2.46
< Back
Install
Cancel
Browse For Folder
Select the folder to install OpenVPN 2.2.2 in:
Program Files
CMAK
Common Files
CONEXANT
DIFX
DisplayLink Core Software
FileZilla FTP Client
Intel
Internet Explorer
Microsoft Office
Microsoft Office 15
Microsoft Policy Platform
Microsoft Silverlight
Make New Folder
OK
Cancel


OpenVPN 2.2.2 Setup
Installation Complete
Setup was completed successfully.
Completed
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVP...
Create shortcut: C:\Users\Public\Desktop\OpenVPN GUI.lnk
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVP...
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVP...
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVP...
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVP...
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVP...
Created uninstaller: C:\Program Files\OpenVPN\Uninstall.exe
Completed
Nullsoft Install System v2.46
< Back
Next >
Cancel


OpenVPN 2.2.2 Setup
Completing the OpenVPN 2.2.2 Setup Wizard
OpenVPN 2.2.2 has been installed on your computer.
Click Finish to close this wizard.
Show Readme
< Back
Finish
Cancel

2.2 Setting up your own Certificate Authority (CA) and generating certificates and keys for an OpenVPN server and multiple clients

The first step in building an OpenVPN 2.x configuration is to establish a PKI (public key infrastructure). The PKI consists of:

- a separate certificate (also known as a public key) and private key for the server and each client
- a master Certificate Authority (CA) certificate and key which is used to sign each of the server and client certificates.

OpenVPN supports bidirectional authentication based on certificates, meaning that the client must authenticate the server certificate and the server must authenticate the client certificate before mutual trust is established.

Both server and client will authenticate the other by first verifying that the presented certificate was signed by the master certificate authority (CA), and then by testing information in the now-authenticated certificate header, such as the certificate common name or certificate type (client or server).

This security model has a number of desirable features from the VPN perspective:

- - The server only needs its own certificate/key -- it doesn't need to know the individual certificates of every client which might possibly connect to it.
- - The server will only accept clients whose certificates were signed by the master CA certificate (which we will generate below). And because the server can perform this signature verification without needing access to the CA private key itself, it is possible for the CA key (the most sensitive key in the entire PKI) to reside on a completely different machine, even one without a network connection.
- - If a private key is compromised, it can be disabled by adding its certificate to a CRL (certificate revocation list). The CRL allows compromised certificates to be selectively rejected without requiring that the entire PKI be rebuilt.
- - The server can enforce client-specific access rights based on embedded certificate fields, such as the Common Name.

Note that the server and client clocks need to be roughly in sync or certificates might not work properly.

2.2.1 Generate the master Certificate Authority (CA) certificate & key

Note: If certificates and key files have already been created, skip to section 3.

In this section we will generate a master CA certificate/key, a server certificate/key, and certificates/keys for the client.

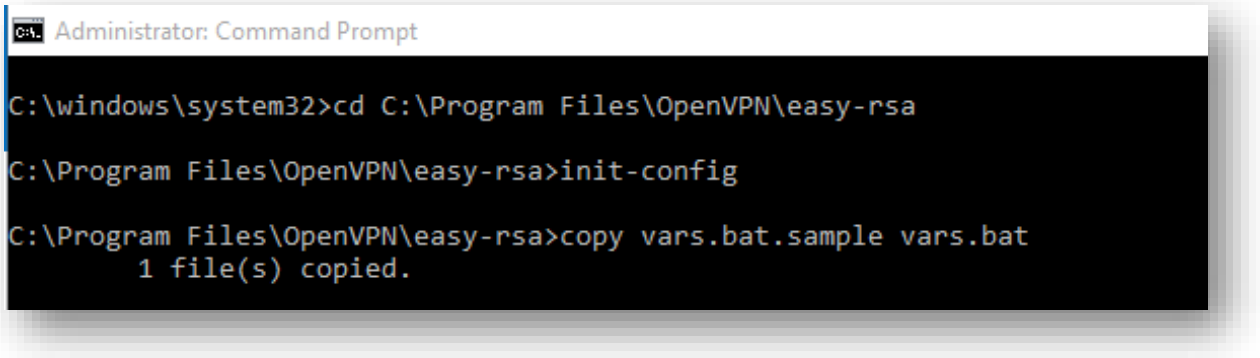
For PKI management, we will use easy-rsa that is included in OpenVPN installation.

On Windows, open up a Command Prompt window and cd to **C:\Program Files\OpenVPN\easy-rsa**

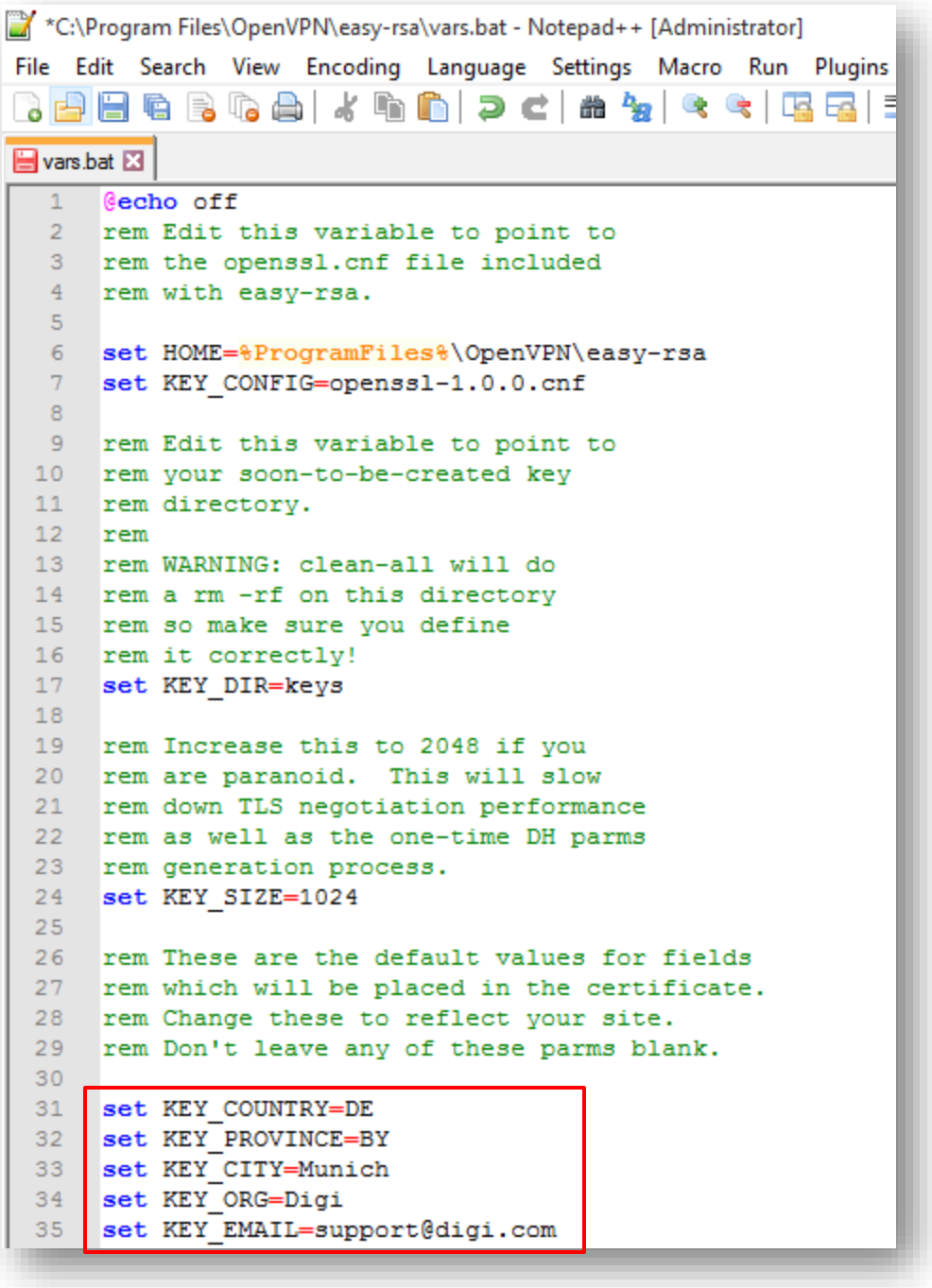
Run the following batch file to copy configuration files into place (this will overwrite any preexisting vars.bat and openssl.cnf files):

```
init-config
```

The output will be like the following:



Now edit the vars file (called vars.bat on Windows) and set the KEY_COUNTRY, KEY_PROVINCE, KEY_CITY, KEY_ORG, and KEY_EMAIL parameters. Don't leave any of these para meters blank:



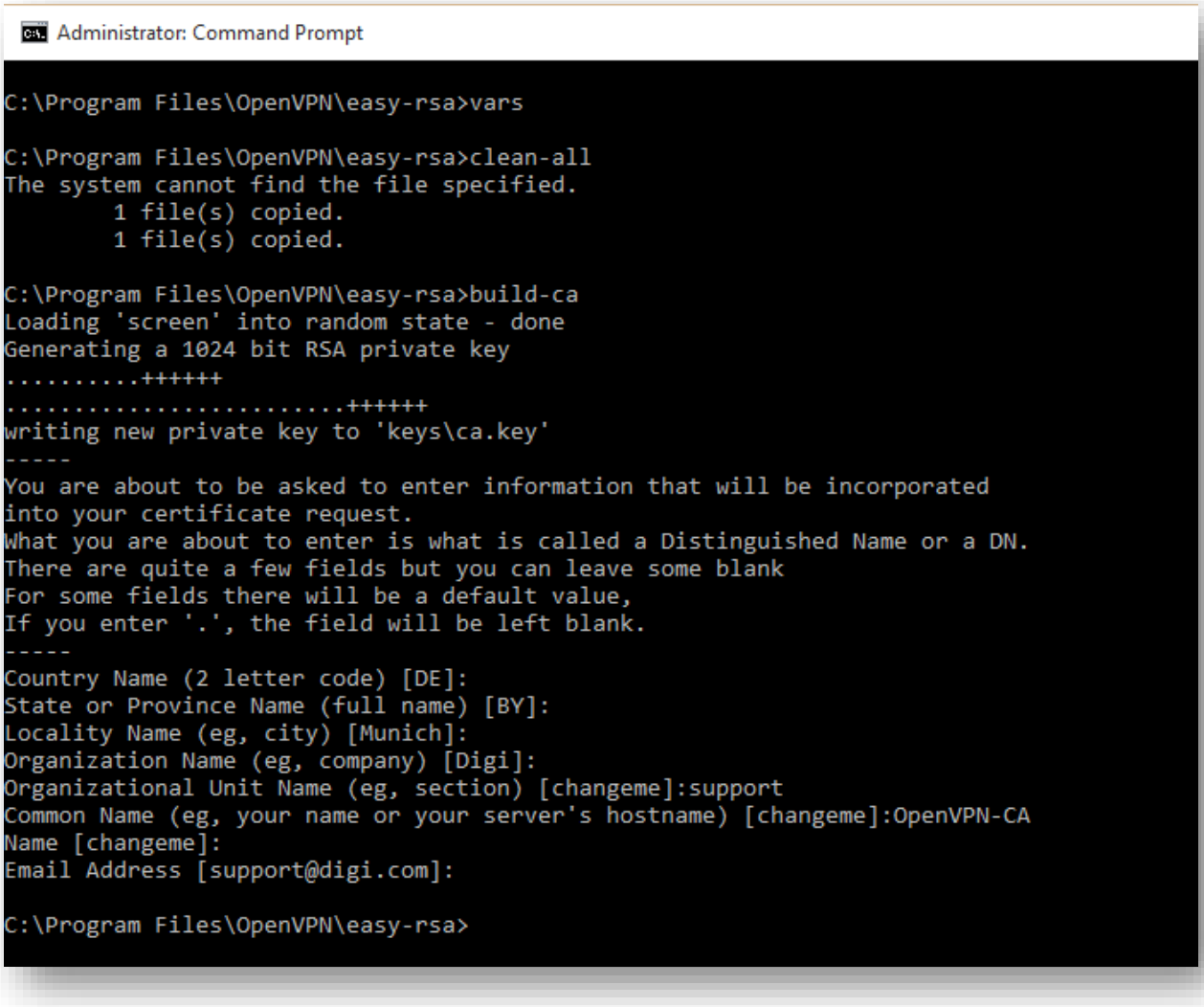
Save and close it.

Then, in command prompt run the following to initialize the PKI:

```
vars
clean-all
build-ca
```

The final command (build-ca) will build the certificate authority (CA) certificate and key by invoking the interactive openssl command.

The output will be like the following:



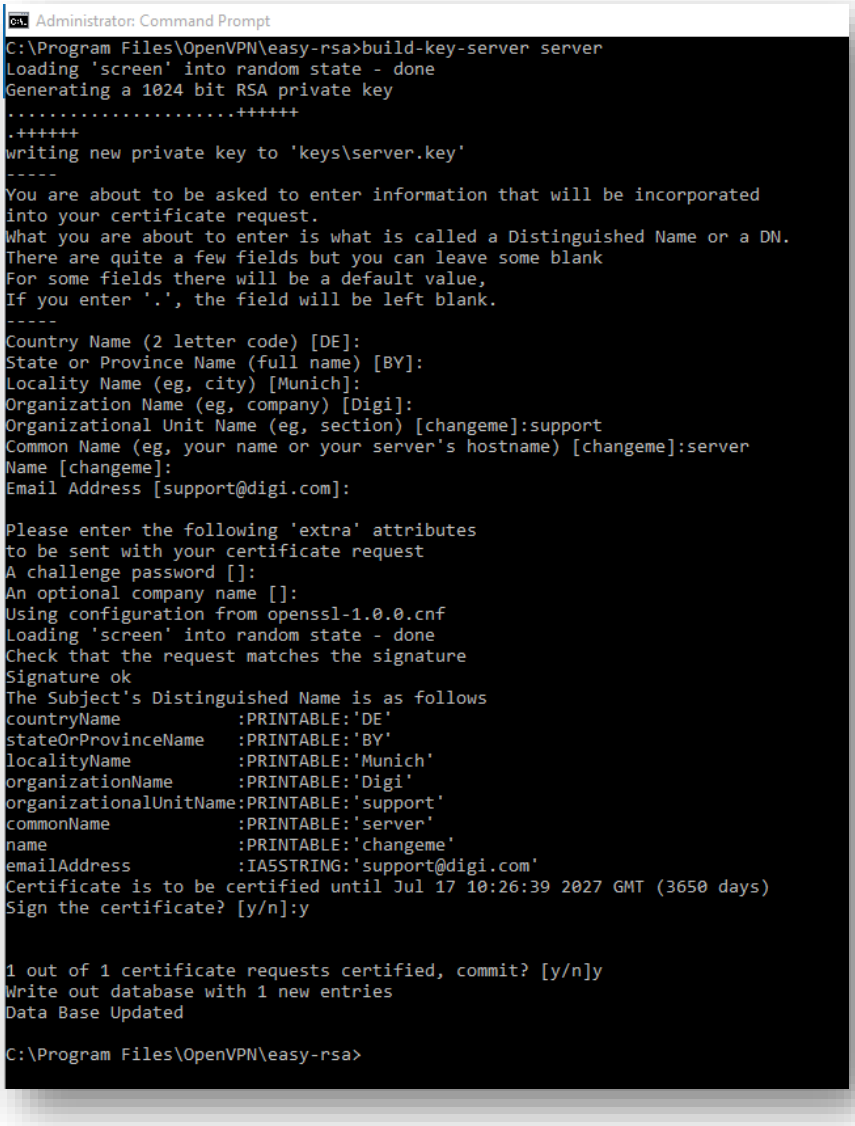
Note that in the above sequence, most queried parameters were defaulted to the values set in the vars or vars.bat files. The only parameter which must be explicitly entered is the Common Name. In the example above, OpenVPN-CA is used

2.2.2 Generate certificate & key for server

Next, we will generate a certificate and private key for the server

```
build-key-server server
```

As in the previous step, most parameters can be defaulted. When the Common Name is queried, enter "server". Two other queries require positive responses, "Sign the certificate? [y/n]" and "1 out of 1 certificate requests certified, commit? [y/n]".



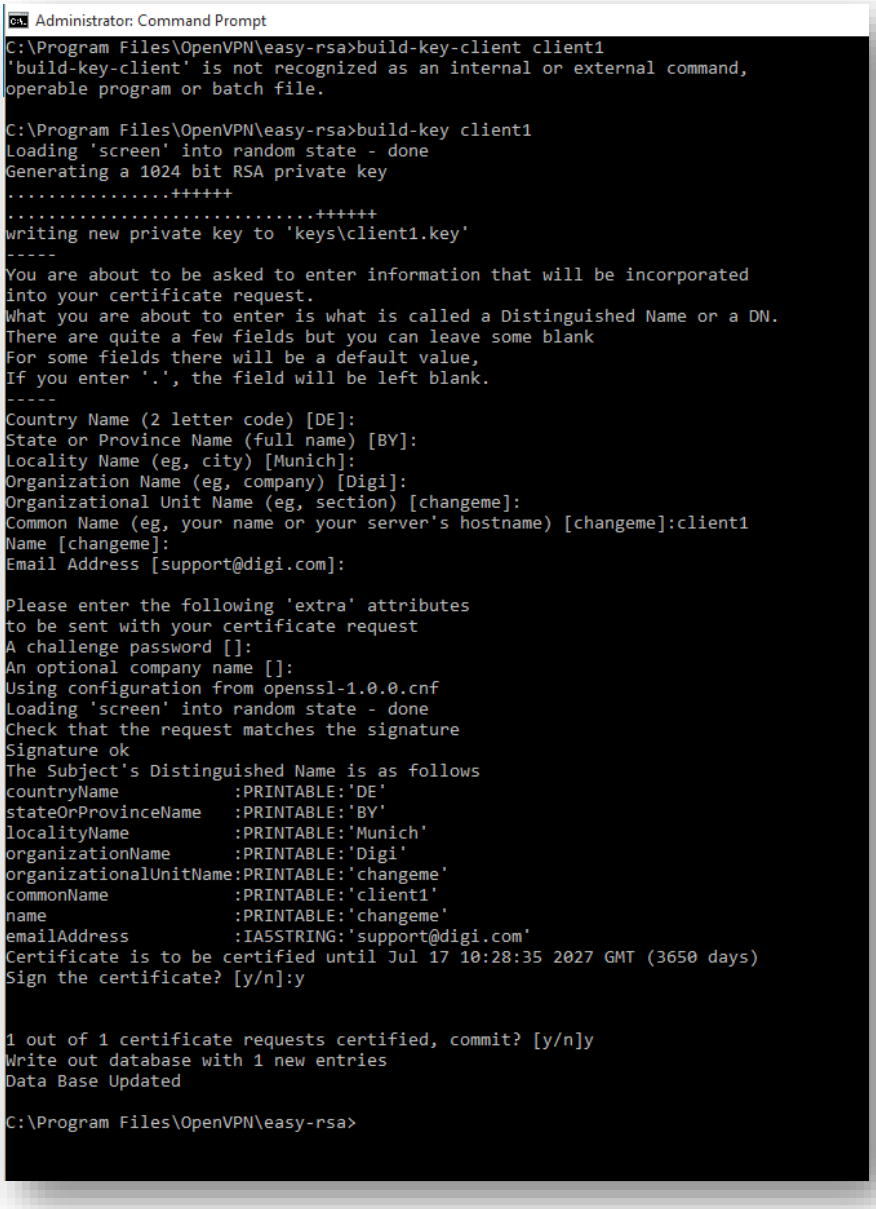
2.2.3 Generate certificates & keys for the 2 clients

Generating client certificates is very similar to the previous step:

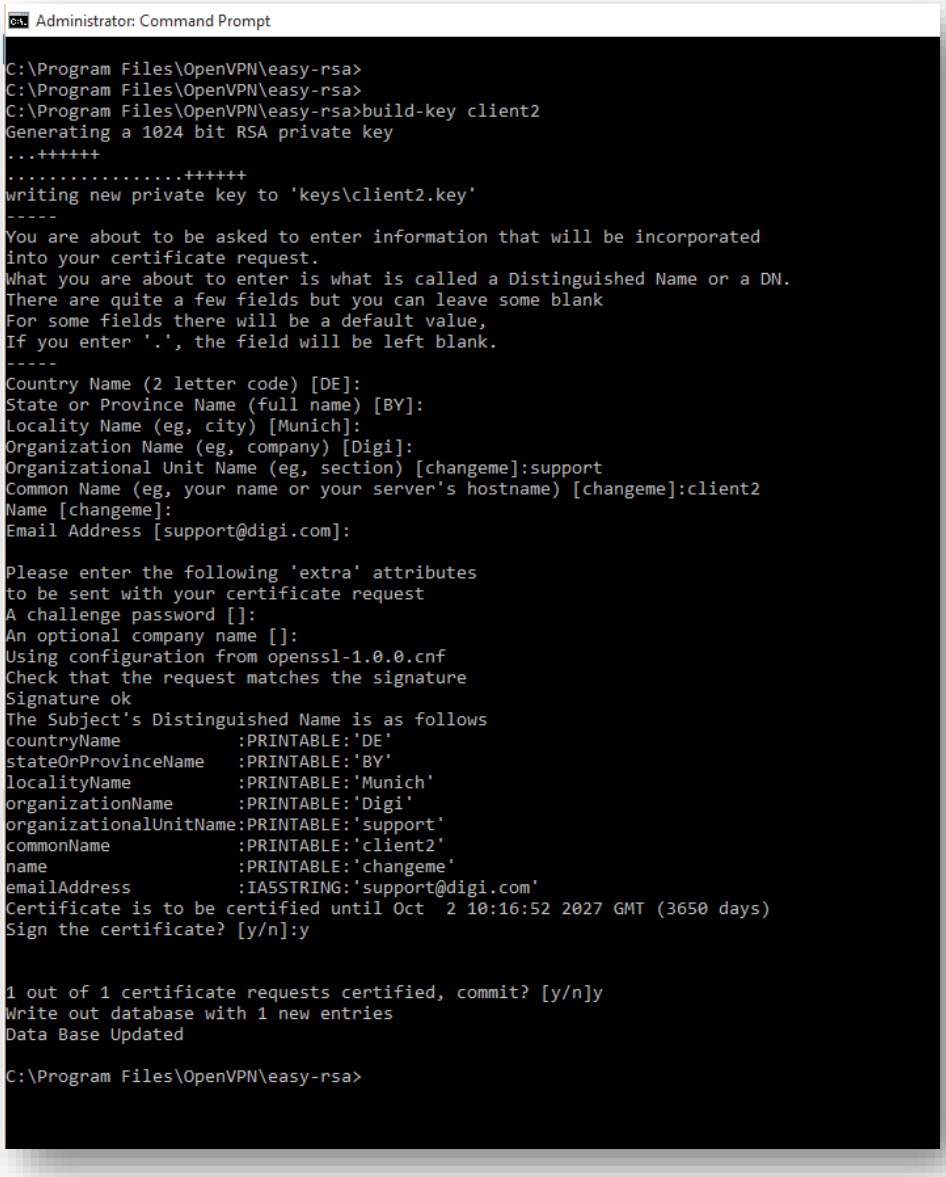
```
build-key client1
build-key client2
```

NOTE: for each client, make sure to type the appropriate Common Name when prompted, i.e. "client1", "client2", or "client3" and always use a unique common name for each client.

Creating client1 certificates:



Creating client2 certificates:



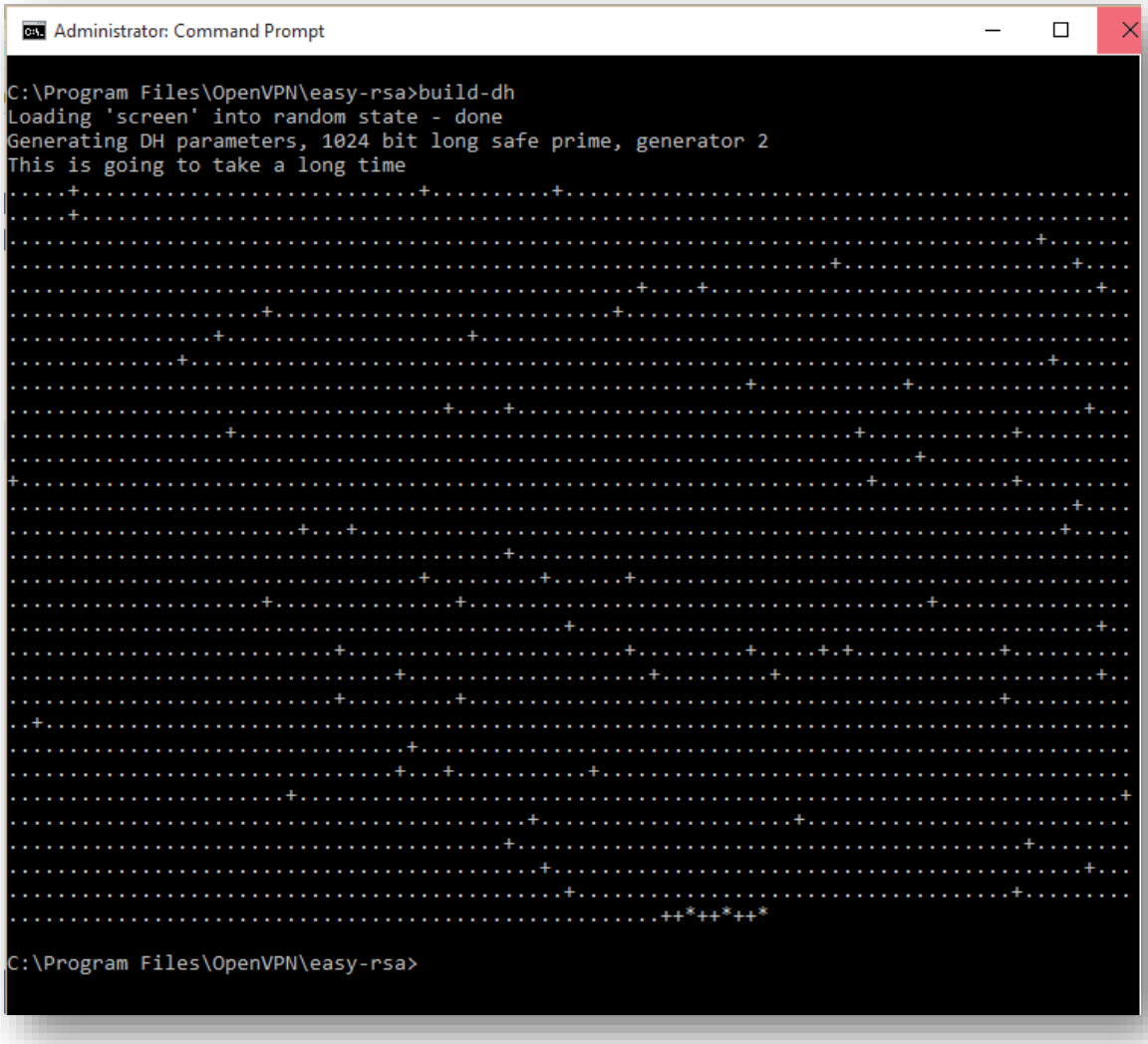
If you would like to password-protect your client keys, substitute the build-key-pass script.

2.2.4 Generate Diffie Hellman parameters

Diffie Hellman parameters must be generated for the OpenVPN server.

On Windows:

```
build-dh
```



2.2.5 Key Files

Now we will find our newly generated keys and certificates in the keys subdirectory. Here is an explanation of the relevant files:

Filename	Needed By	Purpose	Secret
ca.crt	server + all clients	Root CA certificate	NO
ca.key	key signing machine only	Root CA key	YES
dh{n}.pem	server only	Diffie Hellman parameters	NO
server.crt	server only	Server Certificate	NO
server.key	server only	Server Key	YES
client1.crt	client1 only	Client1 Certificate	NO
client1.key	client1 only	Client1 Key	YES
client2.crt	client2 only	Client2 Certificate	NO
client2.key	client2 only	Client2 Key	YES

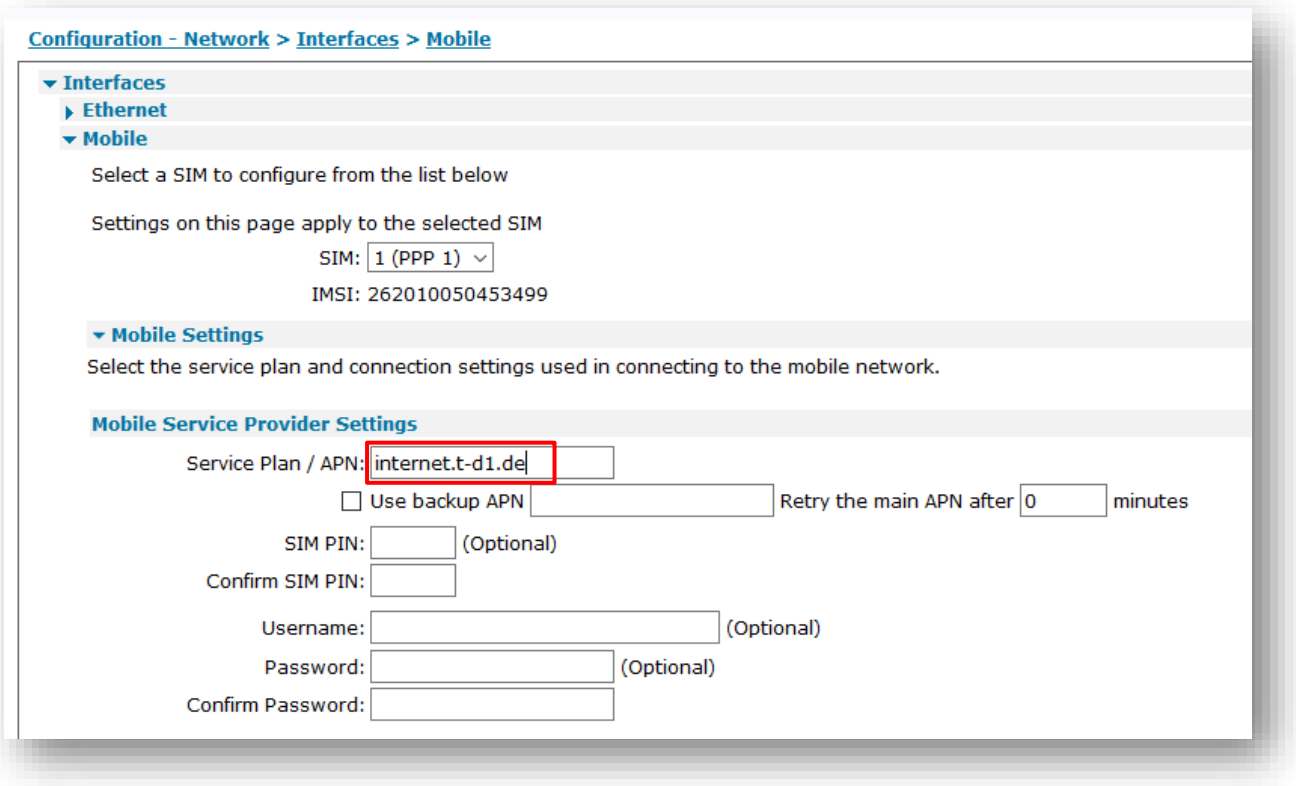
The final step in the key generation process is to copy all files to the WR44 & clients which need them, taking care to copy secret files over a secure channel.

3 TRANSPORT WR CONFIGURATION

3.1 WAN Interface configuration

In this example, the TransPort WR has the Mobile interface as the WAN interface and it is configured as follows:

CONFIGURATION - NETWORK > INTERFACES > MOBILE



Where:

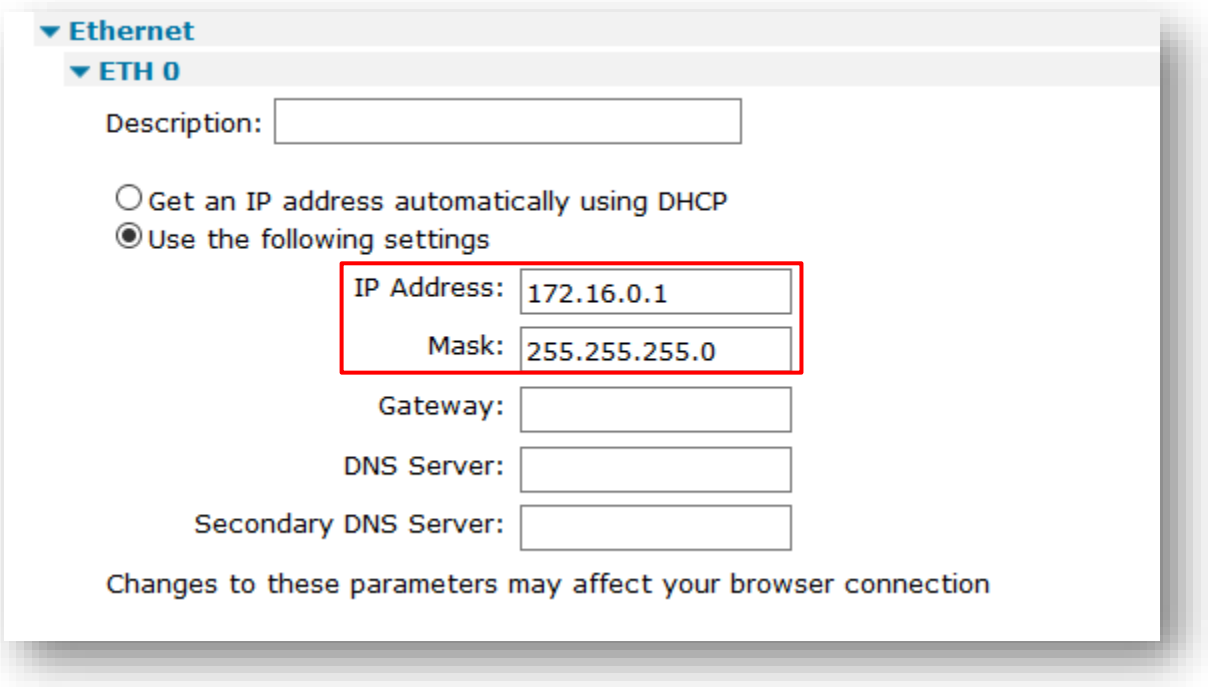
Parameter	Setting	Description
Service Plan/APN	Internet.t-d1.de	Enter the APN of your mobile provider

Please note: Depending on provider, a SIM PIN or Username/Password may be required. If needed, enter them in the appropriate fields.

3.2 LAN Interface configuration

In this example, the LAN interface is configured with a static address as follows:

CONFIGURATION - NETWORK > INTERFACES > ETHERNET > ETH 0



Where:

Parameter	Setting	Description
IP Address	172.16.0.1	Enter the IP address of the LAN interface for the router
Mask	255.255.255.0	Enter the subnet mask

3.3 Transfer Certificates and Key files

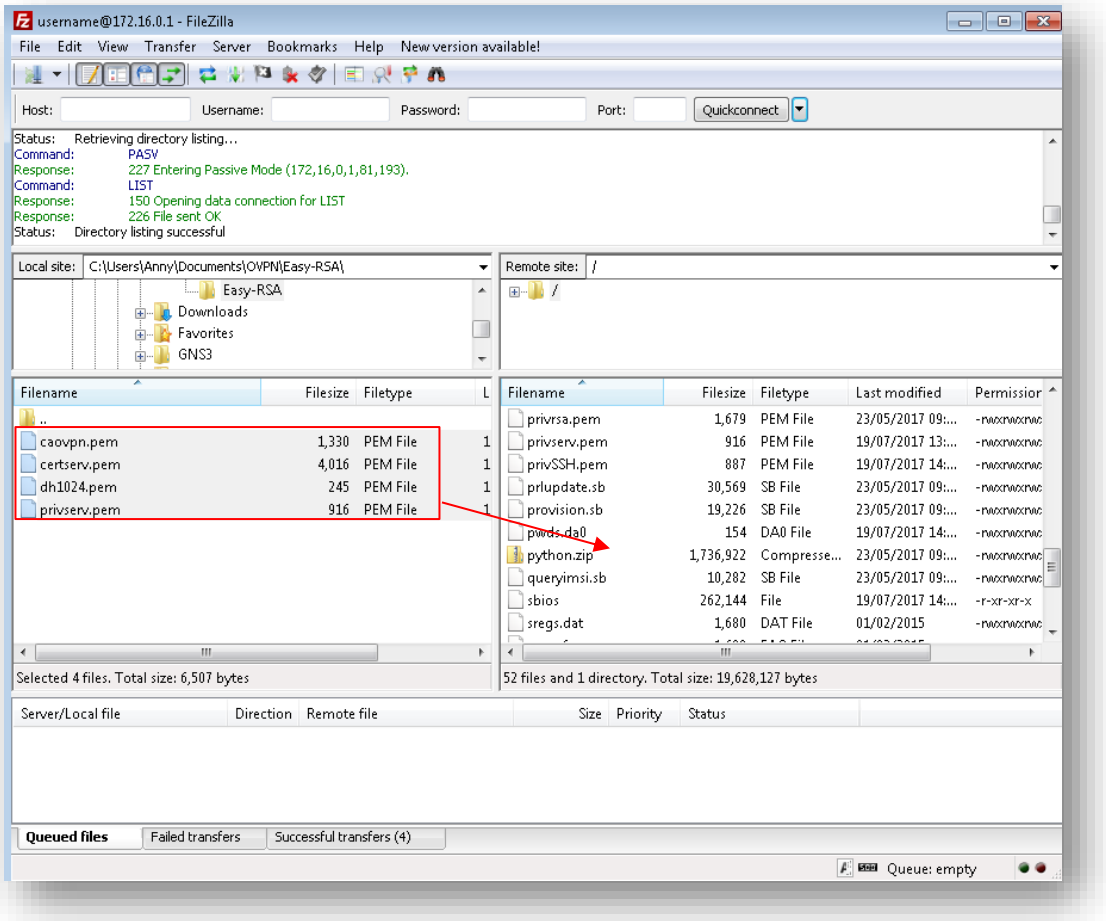
Before to transfer the Certficates and Key files on the server, they must be renamed as follows:

Filename	Purpose	New FileName
ca.crt	Root CA certificate	**ca**ovpn.pem
server.crt	Server Certificate	**cert**serv.pem
server.key	Server Key	**priv**serv.pem

The Diffie Hellman parameters file should remain unchanged.

Once done that, the files can be transferred into the Server using for example an FTP client, connected with the TransPort router with usual username and password.

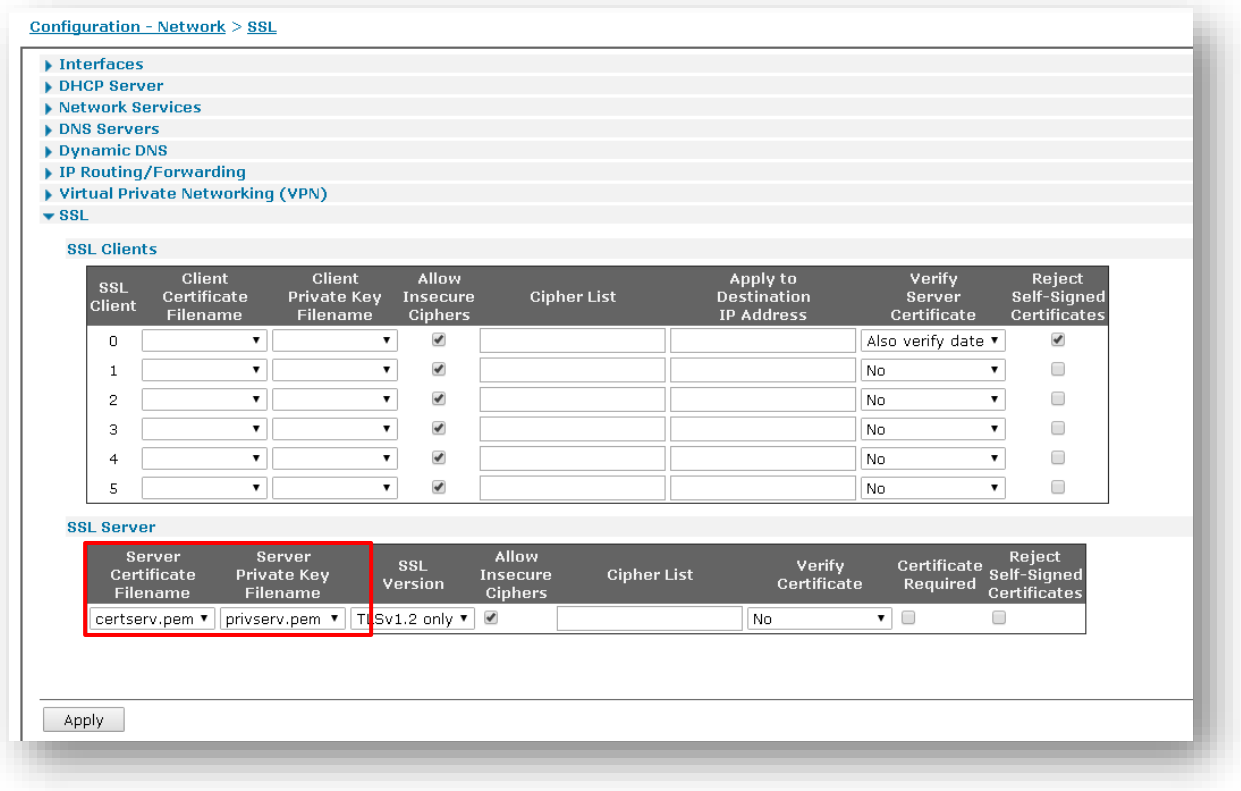
Please note that you may need to change your IP on the laptop accordingly with the new IP address configured on the ETH0 of the router.



3.4 SSL Certificates configuration

When the certificates have been transferred to the Server, the router needs to be configured so it knows which server certificate files to use:

CONFIGURATION – NETWORK > SSL



Where:

Parameter	Setting	Description
Server Certificate Filename	certserv.pem	The file containing the server certificate is selected from this drop-down list. In this example this the one just transferred to the router.
Server Private Key Filename	privserv.pem	The file containing the private key that matches the above certificate is selected from this drop-down list. In this example this the one just transferred to the router.

3.5 OpenVPN Server mode configuration

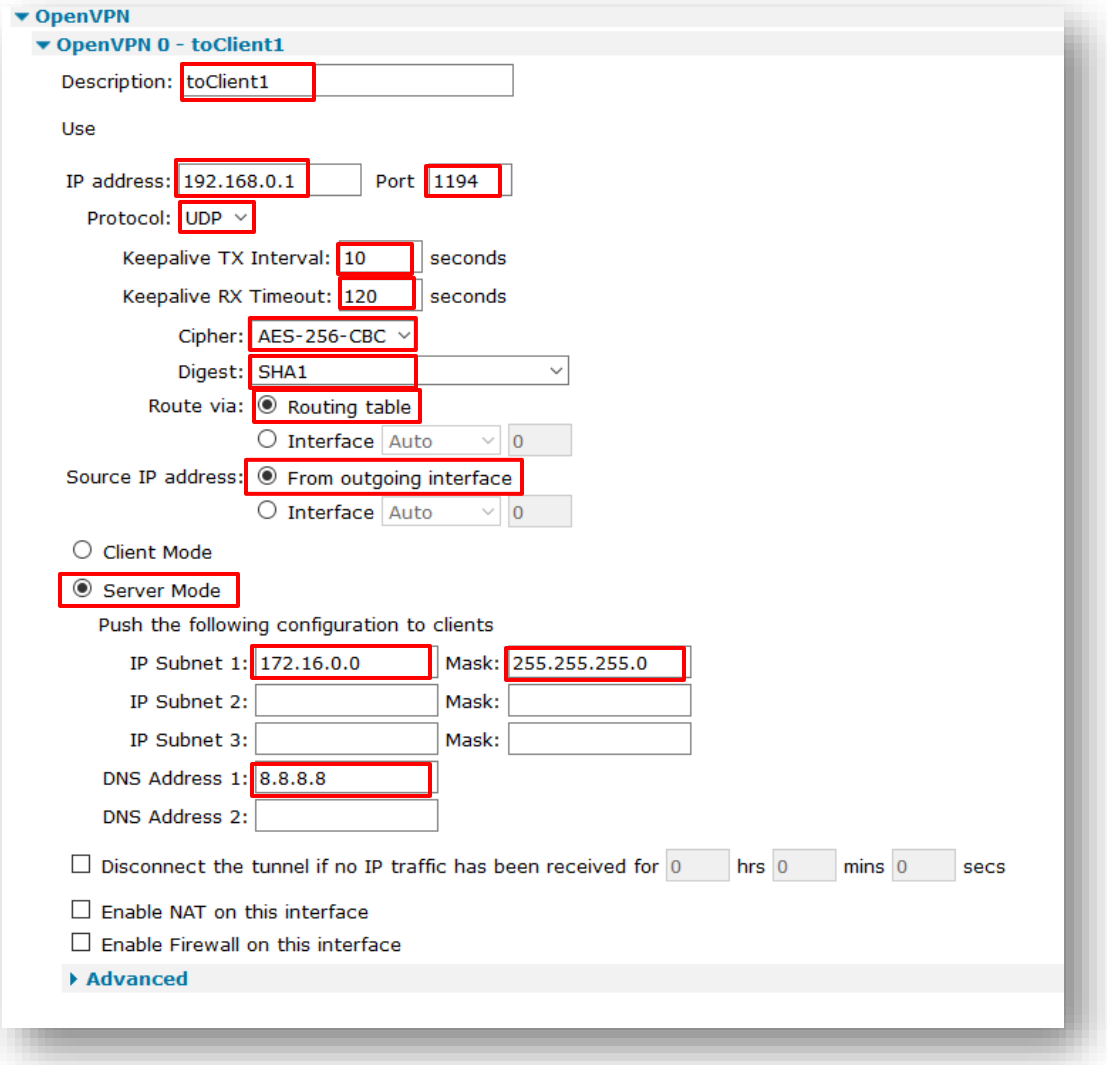
An OpenVPN interface will be configured on the TransPort router that acts as OpenVPN server. There should be as many OpenVPN interfaces configured as the number or required concurrent VPN connections. For example, if there are 10 remote users and there are likely to be 3 connected at any one time, 3 OpenVPN interfaces will be needed.

In case of multiple clients, this is not directly related to either clifilezent1 or client 2. But are a set of parameters that must match and have the correct settings for any client that tries to connect in.

In this Application Note, there is two remote users, so two OpenVPN interfaces will be configured:

3.5.1 OVPN Server Interface and Routing for Client 1

CONFIGURATION - NETWORK > VIRTUAL PRIVATE NETWORKING (VPN) > OPENVPN > OPENVPN 0

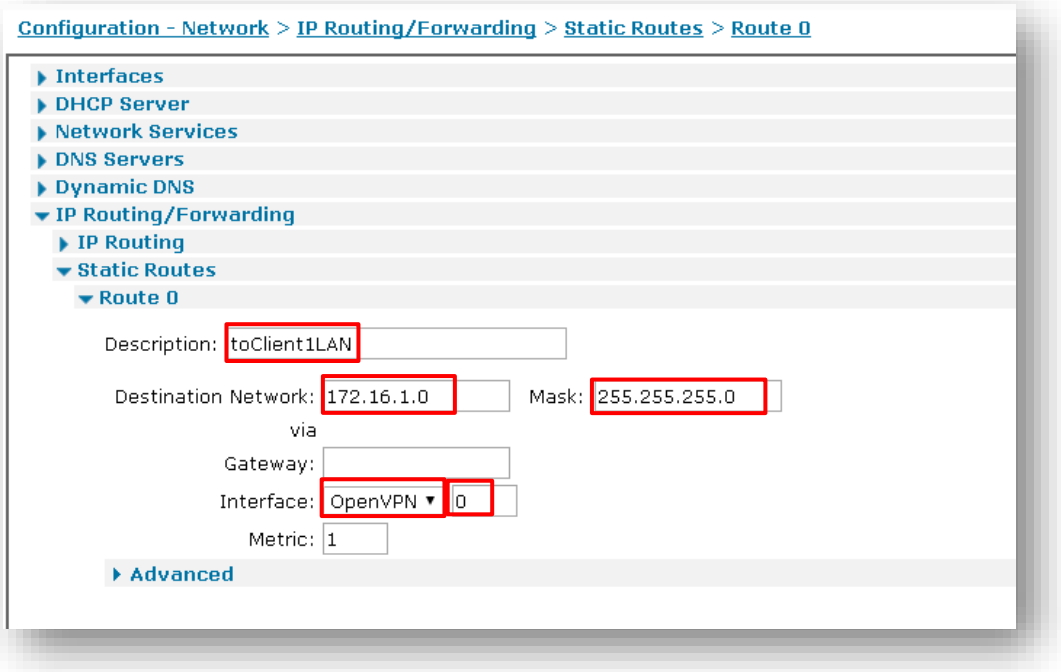


Where:

Parameter	Setting	Description
Description	toClient1	Friendly name for this interface
IP address	192.168.0.1	IP address for this interface. OpenVPN interfaces use a 30 bit mask, the first address is the network address, the 2nd is the server address, the 3rd is the client address, and the 4th is the broadcast address. This address must be configured as the 2nd IP address in the block of 4.
Port	1194 (default)	This is the TCP or UDP port number that the server will listen on for incoming VPN connections
Protocol	UDP (default)	This will either be TCP or UDP. It is up to the reader to decide which protocol to use, both the server and all clients must use the same protocol. See note below with regards to protocol choice.
Keepalive TX Interval	10	Keepalive interval: Interval between OpenVPN ping transmissions. These are required to detect the operational state of the VPN connection.
Keepalive RX Timeout	120	Keepalive timeout before VPN is marked as down: If the server hasn't received a ping from the client in the time limit specified, the tunnel will be marked as down
Cipher	AES-256-CBC	Encryption algorithm to use. The cipher is not negotiated during tunnel establishment. The server and all clients must be configured to use the same cipher. If the ciphers do not match, decryption errors will occur.
Digest	SHA1 (default)	Authentication algorithm to use. The digest is not negotiated during tunnel establishment. The server and all clients must be configured to use the same digest. If the ciphers do not match, authentication errors will occur.
Route via	Routing table (default)	Uses the routing table to determine the best route
Source IP address	From outgoing interface (default)	The IP address of the outgoing interface will be used as the source IP address
Server mode	Selected	Enables server mode.This should be enabled so the OpenVPN interface will answer incoming VPN connections.
Push IP Subnet 1	172.16.0.0	Network IP address to push as a route. These parameters are used to push routing information to the remote VPN client. All subnets that can and must be accessed via the VPN tunnel should be specified here.
Push IP mask 1	255.255.255.0	Network IP mask to push as a route. This is used in conjunction with the IP address field above
DNS Address 1	8.8.8.8	DNS address to push to the client

In order to enable the router to reach the LAN of the client 1, a route must be configured for this subnet, with the outgoing interface being the OVPN 0 one:

CONFIGURATION - NETWORK > IP ROUTING/FORWARDING > STATIC ROUTES > ROUTE 0

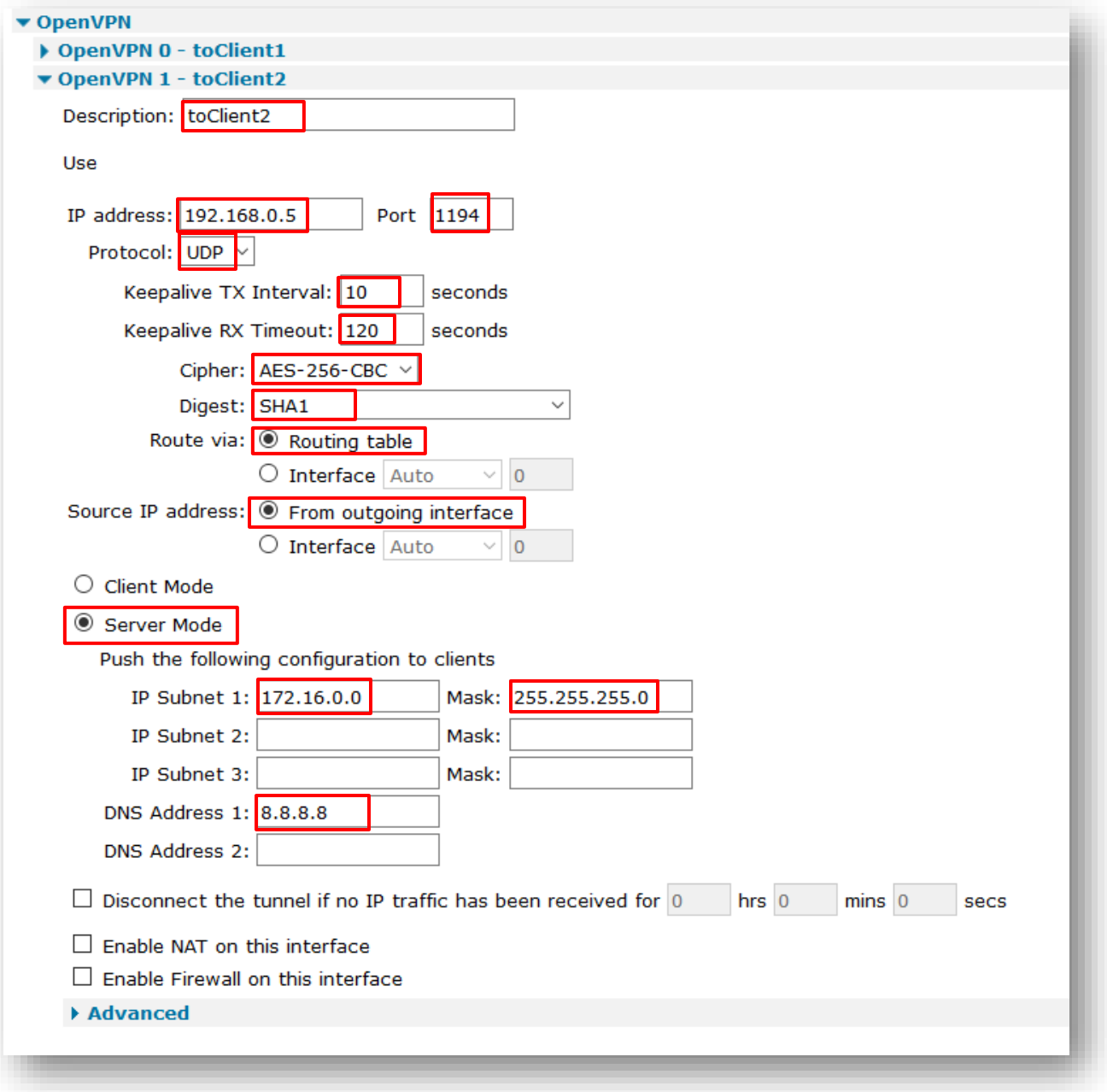


Where:

Parameter	Setting	Description
Description	ToClient1LAN	Friendly name for this static route
Destination Network-Mask	172.16.1.0-255.255.255.0	The IP address of the destination subnet, network or IP address for the route. If the router receives a packet with a destination IP address that matches the Destination Network/Mask combination it will route the packet through the interface specified below. In this example, the destination subnet is the Client one.
Interface	OpenVPN 0	The interface for routing the packets. Select from the drop-down list and enter the interface instance number in the adjacent text box. In this example, this is the OVPN interface just configured.

3.5.2 OVPN Server Interface and Routing for Client 2

CONFIGURATION - NETWORK > VIRTUAL PRIVATE NETWORKING (VPN) > OPENVPN > OPENVPN 1

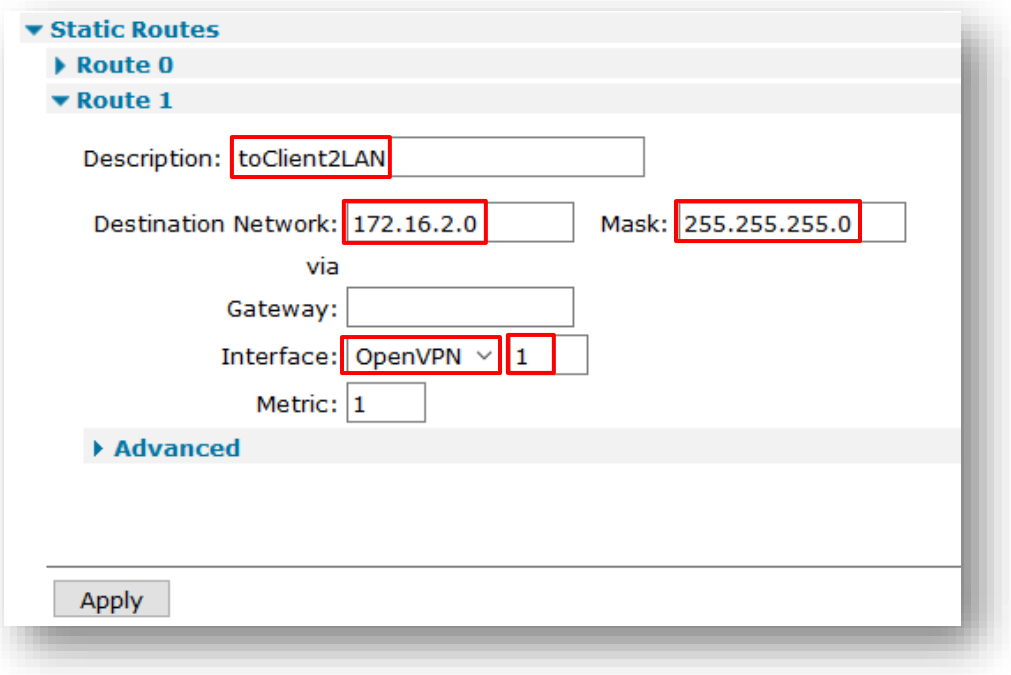


Where:

Parameter	Setting	Description
Description	toClient2	Friendly name for this interface
IP address	192.168.0.5	IP address for this interface. OpenVPN interfaces use a 30 bit mask, the first address is the network address, the 2nd is the server address, the 3rd is the client address, and the 4th is the broadcast address. This address must be configured as the 2nd IP address in the block of 4.
Port	1194 (default)	This is the TCP or UDP port number that the server will listen on for incoming VPN connections
Protocol	UDP (default)	This will either be TCP or UDP. It is up to the reader to decide which protocol to use, both the server and all clients must use the same protocol. See note below with regards to protocol choice.
Keepalive TX Interval	10	Keepalive interval: Interval between OpenVPN ping transmissions. These are required to detect the operational state of the VPN connection.
Keepalive RX Timeout	120	Keepalive timeout before VPN is marked as down: If the server hasn't received a ping from the client in the time limit specified, the tunnel will be marked as down
Cipher	AES-256-CBC	Encryption algorithm to use. The cipher is not negotiated during tunnel establishment. The server and all clients must be configured to use the same cipher. If the ciphers do not match, decryption errors will occur.
Digest	SHA1 (default)	Authentication algorithm to use. The digest is not negotiated during tunnel establishment. The server and all clients must be configured to use the same digest. If the ciphers do not match, authentication errors will occur.
Route via	Routing table (default)	Uses the routing table to determine the best route
Source IP address	From outgoing interface (default)	The IP address of the outgoing interface will be used as the source IP address
Server mode	Selected	Enables server mode.This should be enabled so the OpenVPN interface will answer incoming VPN connections.
Push IP Subnet 1	172.16.0.0	Network IP address to push as a route. These parameters are used to push routing information to the remote VPN client. All subnets that can and must be accessed via the VPN tunnel should be specified here.
Push IP mask 1	255.255.255.0	Network IP mask to push as a route. This is used in conjunction with the IP address field above
DNS Address 1	8.8.8.8	DNS address to push to the client

In order to enable the router to reach the LAN of the client 2, a route must be configured for this subnet, with the outgoing interface being the OVPN 1 one:

CONFIGURATION - NETWORK > IP ROUTING/FORWARDING > STATIC ROUTES > ROUTE 1



Where:

Parameter	Setting	Description
Description	ToClient2LAN	Friendly name for this static route
Destination Network-Mask	172.16.2.0-255.255.255.0	The IP address of the destination subnet, network or IP address for the route. If the router receives a packet with a destination IP address that matches the Destination Network/Mask combination it will route the packet through the interface specified below. In this example, the destination subnet is the Client one.
Interface	OpenVPN 1	The interface for routing the packets. Select from the drop-down list and enter the interface instance number in the adjacent text box. In this example, this is the OVPN interface just configured.

3.5.3 Note regarding TCP or UDP

UDP has less protocol overhead than TCP as there is no reliability support built into UDP. A data channel packet (a packet to be tunnelled) gets encrypted and set as the payload of a UDP packet before being sent on its way. If the packet is dropped, no retransmissions of the encrypted packet will occur. It is up to the higher layers to detect that a packet has been lost and go about retransmitting. It is more difficult to detect that a peer has disconnected though, and no indication is sent to the peer if the local end closes the socket. For that reason use of OpenVPN pings is generally required to confirm that the tunnel is still established. If no pings are received within a period of time the tunnel should be deemed to be failed and the tunnel should be torn down. A reliability layer is built into OpenVPN to ensure that control channel packets are transmitted to the remote peer. This reliability layer is used whether using TCP or UDP for the link transport.

TCP has higher overhead than UDP as all data is acknowledged. Also, there are issues that cause problems when transporting TCP traffic over a TCP link. This is effectively what will be occurring when a TCP stream is tunnelled through an OpenVPN tunnel configured to use TCP as the transport layer. Data transfer can get quite bogged down when retransmits start occurring. With TCP as the link transport protocol however, all traffic will get through the tunnel with no packet loss at all. When using TCP, it is much clearer when a socket has been closed by the other peer. Notifications will be delivered to the OpenVPN task that the socket has closed in a timely fashion without the need to rely on traffic through the tunnel. For this reason, there is less need to configure the peers to deliver OpenVPN pings through the data channel to confirm connectivity. With TCP, TCP keepalives can be used to keep the underlying interface connected. The bottom line is that less traffic needs to flow to confirm tunnel connectivity during times of low traffic through the tunnel.

4 CLIENT CONFIGURATION

The following steps explain the configuration that needs to be done on the 2 remote user's laptops.

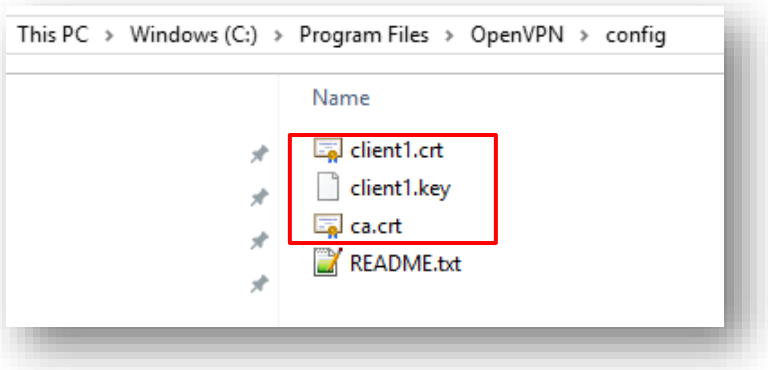
4.1 Install the OpenVPN software

Using the same installation package that was downloaded earlier, install OpenVPN in exactly the same manner as before and selecting the same options. See section 2.1 for screen shots and instructions.

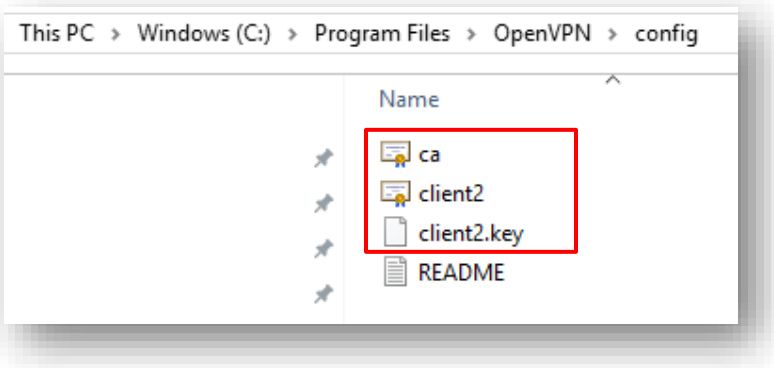
4.2 Install the SSL certificates

The SSL certificates that were created earlier should now be securely transferred onto the two users laptops from the Certificate Authority PC. The files should be placed on both laptop in the directory *C:\Program Files\OpenVPN\config:*

Client1:

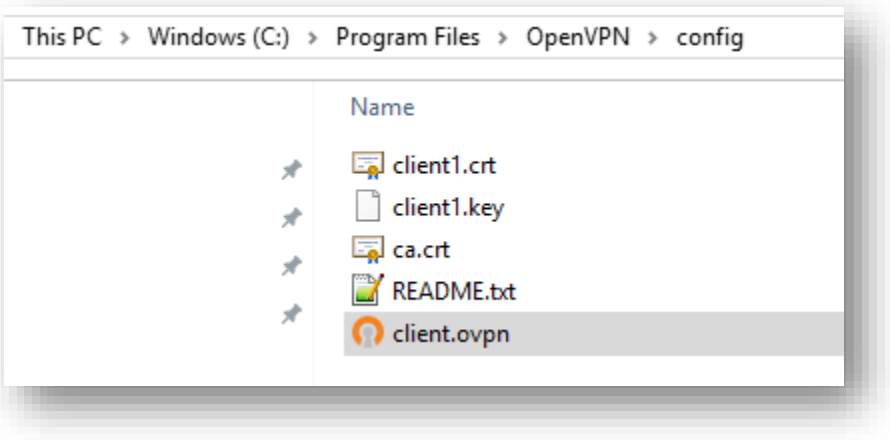


Client2:



4.3 Windows OpenVPN Client 1 configuration

Copy the sample client config (client.ovpn) from *C:\Program Files\OpenVPN\sample-config* to the main config directory where the certificates are located *C:\Program Files\OpenVPN\config:*



Open and edit the client.ovpn file using notepad

Take note of the parts in red! These lines are the most important ones and some have been changed from the sample config defaults.

Extra comments have been added in blue.

```
##############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
##############################################

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
```

```
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap

# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 10.104.1.115 1194
;remote my-server-2 1194

# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nobody

# Try to preserve some state across restarts.
persist-key
persist-tun

# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here.  See the man page
# if your proxy server requires
# authentication.
```

```
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets.  Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings

# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
# These are the names of the private key and
# certificate files in the config directory
ca ca.crt
cert client1.crt
key client1.key

# Verify server certificate by checking that the
# certicate has the correct key usage set.
# This is an important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the keyUsage set to
#   digitalSignature, keyEncipherment
# and the extendedKeyUsage to
#   serverAuth
# EasyRSA can do this for you.
remote-cert-tls server

# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
# Note that 2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
cipher AES-256-CBC


# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
# Compression MUST BE DISABLED
;comp-lzo

# Set log file verbosity.
```

```
verb 3

# This whole section has been added and is important
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# Silence repeating messages
;mute 20
```

Save and close this file. This user's laptop configuration for Client 1 is now complete.

4.3.1 Additional configuration option for OpenVPN Client version 2.4 and newer

OpenVPN 2.4 and newer limits the default cipher list more than earlier versions did. As this change is not backward compatible, this will prevent the client to connect to the Digi, which uses older ciphers.

When using OpenVPN Client version 2.4 and newer, it is required to add an additional configuration parameter to the **client.ovpn** file.

At the end of the file, add the following:

```
# This whole section has been added and is important
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# Silence repeating messages
;mute 20

#Enable default cipher list for backward compatibility
tls-cipher DEFAULT
```

Save and close this file. This user's laptop configuration for Client 1 is now complete.

4.4 Windows OpenVPN Client 2 configuration

As done for the Client 1, copy the sample client config (client.ovpn) from *C:\Program Files\OpenVPN\sample-config* to the main config directory where the certificates are located *C:\Program Files\OpenVPN\config and edit the* the client.ovpn file using notepad.

The configuration will be similar to the Client 1 one, but the correct certificates and key filenames need to be used.

Take note of the parts in red! These lines are the most important ones and some have been changed from the sample config defaults.

Extra comments have been added in blue

```
##############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
##############################################

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap

# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp

# The hostname/IP and port of the server.
```

```
# You can have multiple remote entries
# to load balance between the servers.
remote 10.104.1.115 1194
;remote my-server-2 1194

# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nobody

# Try to preserve some state across restarts.
persist-key
persist-tun

# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here.  See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets.  Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings

# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca ca.crt
cert client2.crt
key client2.key

# Verify server certificate by checking that the
# certicate has the correct key usage set.
```

```
# This is an important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the keyUsage set to
#   digitalSignature, keyEncipherment
# and the extendedKeyUsage to
#   serverAuth
# EasyRSA can do this for you.
remote-cert-tls server

# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
# Note that 2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
cipher AES-256-CBC


# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
;comp-lzo

# Set log file verbosity.
verb 3

# This whole section has been added and is important
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120


# Silence repeating messages
;mute 20
```

Save and close this file. The user's laptop configuration for Client 2 is now complete.

4.4.1 Additional configuration option for OpenVPN Client version 2.4 and newer

OpenVPN 2.4 and newer limits the default cipher list more than earlier versions did. As this change is not backward compatible, this will prevent the client to connect to the Digi, which uses older ciphers.

When using OpenVPN Client version 2.4 and newer, it is required to add an additional configuration parameter to the **client.ovpn** file.

At the end of the file, add the following:

```
# This whole section has been added and is important
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# Silence repeating messages
;mute 20

#Enable default cipher list for backward compatibility
tls-cipher DEFAULT
```

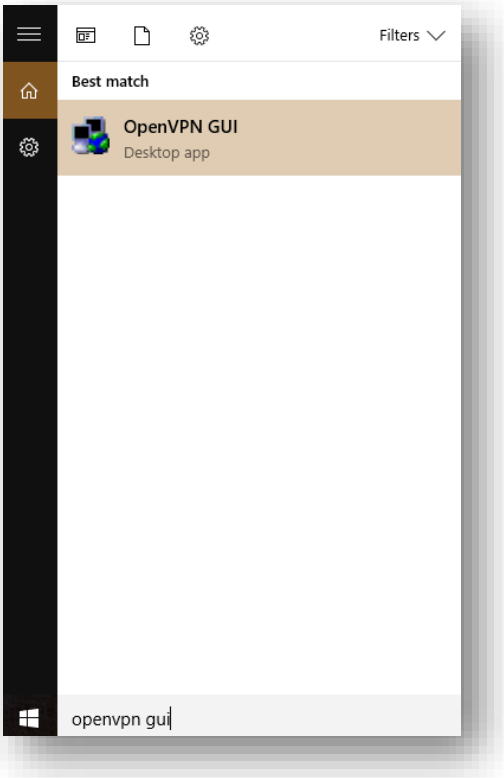
Save and close this file. This user's laptop configuration for Client 2 is now complete.

5 VERIFY CONNECTION DETAILS

5.1 Check OpenVPN connection for Client 1

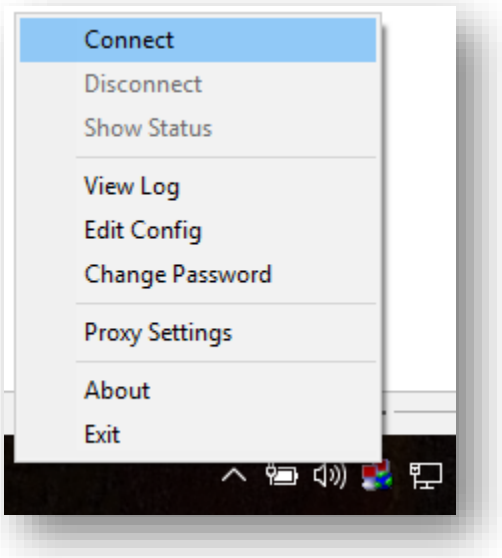
5.1.1 Connect the Client 1

To test the OpenVPN connection, run the OpenVPN software from the Start menu or the desktop shortcut on Client 1 laptop:

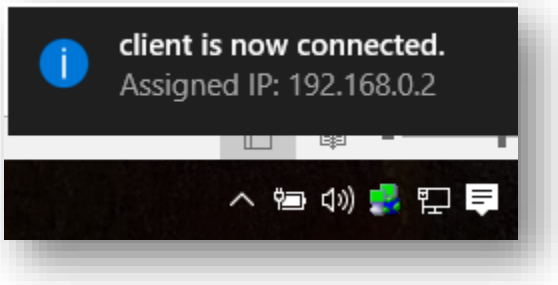


This will run the OpenVPN client software and place the icon in the system tray:

To connect, simply double click the system tray icon or right click and select “connect”:



When the OpenVPN connection is established, the icon will turn green and a notification of the assigned IP address will be shown:



The Client is connected and the IP address assigned is the one configured for the OpenVPN interface OVPN0 on the TransPort WR.

5.1.2 Check Routing Table

Check the routing table for pushed routing information, this should match the network entered into the OpenVPN0 'Push IP address' & 'Push Mask' parameters, only the lines relating to OpenVPN routing are shown below:

```
C:\windows\system32>route print
---SOME LINES REMOVED---

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      10.104.1.1     10.104.1.122     20
       10.104.1.0    255.255.255.0         On-link      10.104.1.122    276
     10.104.1.122  255.255.255.255         On-link      10.104.1.122    276
     10.104.1.255  255.255.255.255         On-link      10.104.1.122    276
       172.16.0.0    255.255.255.0     192.168.0.1      192.168.0.2     30
      192.168.0.0  255.255.255.252         On-link       192.168.0.2    286
      192.168.0.2  255.255.255.255         On-link       192.168.0.2    286
      192.168.0.3  255.255.255.255         On-link       192.168.0.2    286
        224.0.0.0        240.0.0.0         On-link      10.104.1.122    276
        224.0.0.0        240.0.0.0         On-link       192.168.0.2    286
  255.255.255.255  255.255.255.255         On-link      10.104.1.122    276
  255.255.255.255  255.255.255.255         On-link       192.168.0.2    286
===========================================================================
```

The network destination 172.16.0.0 with mask 255.255.255.0 is the route that has been pushed from the OpenVPN server (the WR21).

5.1.3 Check Traffic through the OpenVPN Connection

Ping the LAN interface of the TransPort WR:

```
C:\windows\system32>ping 172.16.0.1

Pinging 172.16.0.1 with 32 bytes of data:
Reply from 172.16.0.1: bytes=32 time=2ms TTL=250
Reply from 172.16.0.1: bytes=32 time=2ms TTL=250
Reply from 172.16.0.1: bytes=32 time=2ms TTL=250
Reply from 172.16.0.1: bytes=32 time=2ms TTL=250
```

Ping the server on the corporate LAN, 172.16.0.100:

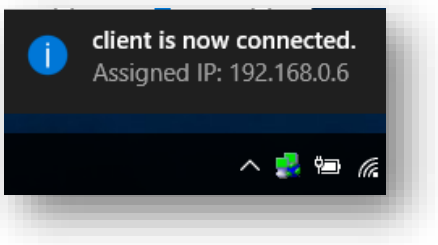
```
C:\windows\system32>ping 172.16.0.100

Pinging 172.16.0.100 with 32 bytes of data:
Reply from 172.16.0.100: bytes=32 time=2ms TTL=127
Reply from 172.16.0.100: bytes=32 time=3ms TTL=127
Reply from 172.16.0.100: bytes=32 time=2ms TTL=127
Reply from 172.16.0.100: bytes=32 time=2ms TTL=127
```

5.2 Check OpenVPN connection for Client 2

5.2.1 Connect the Client 2

To test the OpenVPN connection, run the OpenVPN software from the Start menu or the desktop shortcut on Client 2 laptop following same steps as section 5.1.1.

The second client will connect to the Server getting the IP address configured for the OVPN interface1 (so 192.168.0.6):



The Client is connected and the IP address assigned is the one configured for the OpenVPN interface OVPN1 on the TransPort WR.

5.2.2 Check Routing Table

Check the routing table for pushed routing information, this should match the network entered into the OpenVPN1 'Push IP address' & 'Push Mask' parameters, only the lines relating to OpenVPN routing are shown below:

```
C:\Users\INGTest>route print
---SOME LINES REMOVED---

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      10.104.34.1    10.104.34.111     25
      10.104.34.0    255.255.255.0         On-link     10.104.34.111    281
    10.104.34.111  255.255.255.255         On-link     10.104.34.111    281
    10.104.34.255  255.255.255.255         On-link     10.104.34.111    281
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    306
      172.16.0.0    255.255.255.0      192.168.0.5      192.168.0.6     30
      192.168.0.4  255.255.255.252         On-link       192.168.0.6    286
```

```
      192.168.0.6  255.255.255.255         On-link       192.168.0.6    286
      192.168.0.7  255.255.255.255         On-link       192.168.0.6    286
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link       192.168.0.6    286
        224.0.0.0        240.0.0.0         On-link     10.104.34.111    281
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link       192.168.0.6    286
  255.255.255.255  255.255.255.255         On-link     10.104.34.111    281
===========================================================================
```

5.2.3 Check Traffic through the OpenVPN Connection

Ping the LAN interface of the TransPort WR:

```
C:\Users\INGTest>ping 172.16.0.1

Pinging 172.16.0.1 with 32 bytes of data:
Reply from 172.16.0.1: bytes=32 time=3ms TTL=250
Reply from 172.16.0.1: bytes=32 time=5ms TTL=250
Reply from 172.16.0.1: bytes=32 time=3ms TTL=250
Reply from 172.16.0.1: bytes=32 time=3ms TTL=250

Ping statistics for 172.16.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 5ms, Average = 3ms
```

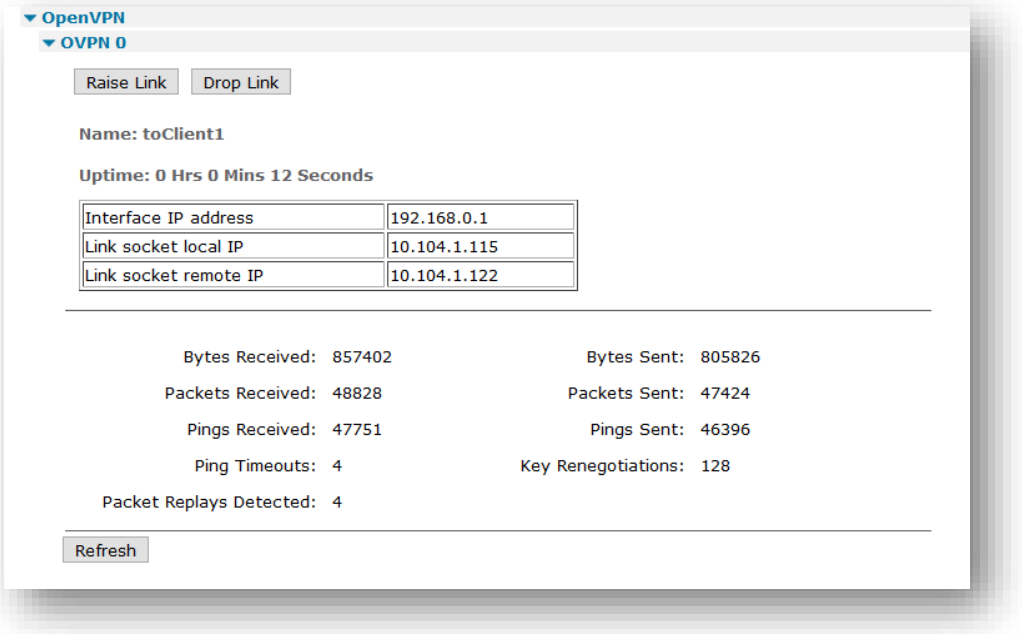
Ping the server on the corporate LAN, 172.16.0.100:

```
C:\Users\INGTest>ping 172.16.0.100

Pinging 172.16.0.100 with 32 bytes of data:
Reply from 172.16.0.100: bytes=32 time=5ms TTL=127
Reply from 172.16.0.100: bytes=32 time=3ms TTL=127
Reply from 172.16.0.100: bytes=32 time=3ms TTL=127
Reply from 172.16.0.100: bytes=32 time=3ms TTL=127

Ping statistics for 172.16.0.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 5ms, Average = 3ms

C:\Users\INGTest>
```

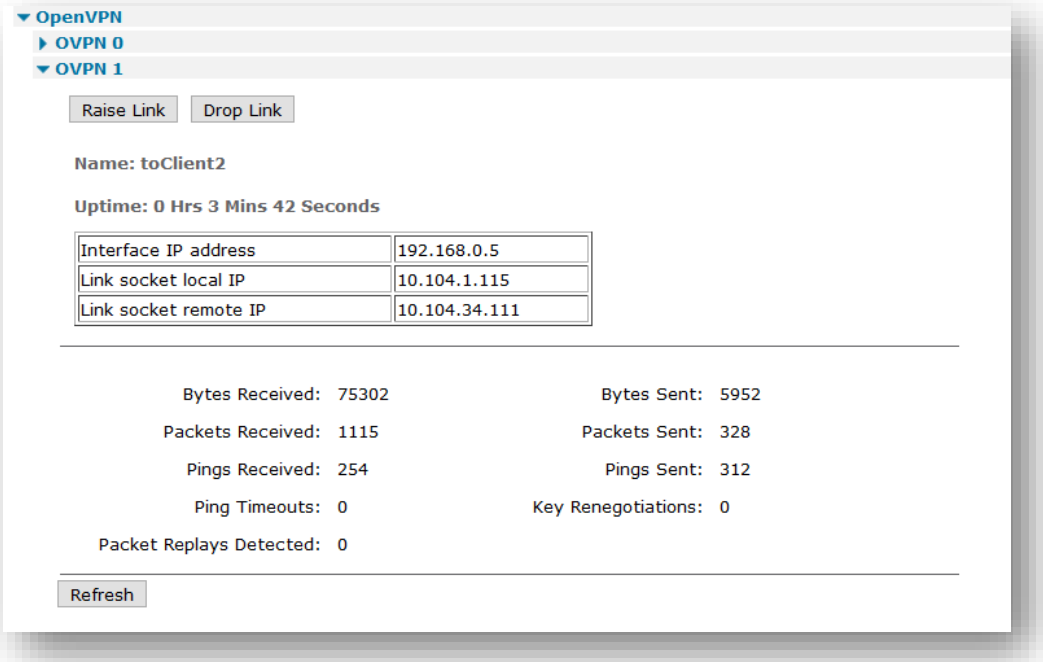
5.3 Check Client 1 and Client 2 OpenVPN Connection from TransPort WR

The VPN status can also be confirmed on the TransPort WR by browsing to:

MANAGEMENT - CONNECTIONS > VIRTUAL PRIVATE NETWORKING (VPN) > OPENVPN > OVPN 0

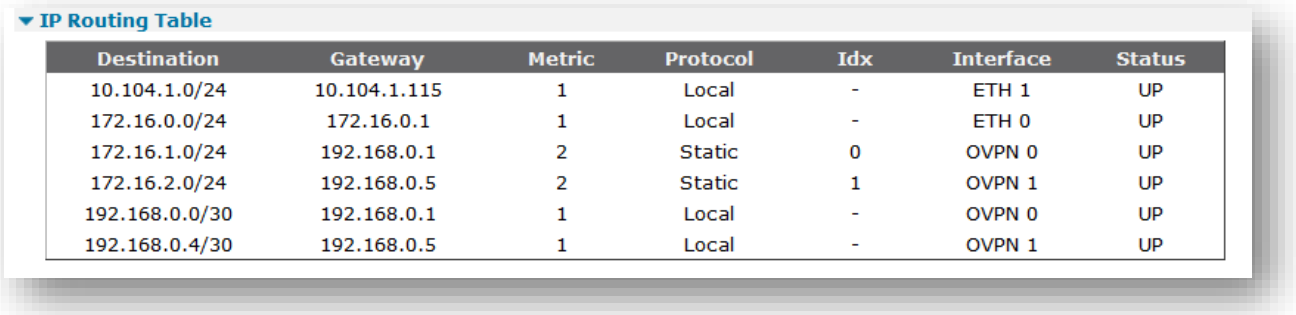


MANAGEMENT - CONNECTIONS > VIRTUAL PRIVATE NETWORKING (VPN) > OPENVPN > OVPN 1



Also the routing table will show the OpenVPN connections with the 2 Clients and the static routes to reach their LANs:

MANAGEMENT - NETWORK STATUS > IP ROUTING TABLE



IP Routing Table

Destination	Gateway	Metric	Protocol	Idx	Interface	Status
10.104.1.0/24	10.104.1.115	1	Local	-	ETH 1	UP
172.16.0.0/24	172.16.0.1	1	Local	-	ETH 0	UP
172.16.1.0/24	192.168.0.1	2	Static	0	OVPN 0	UP
172.16.2.0/24	192.168.0.5	2	Static	1	OVPN 1	UP
192.168.0.0/30	192.168.0.1	1	Local	-	OVPN 0	UP
192.168.0.4/30	192.168.0.5	1	Local	-	OVPN 1	UP

6 REVOKING A CERTIFICATE

Revoking a certificate means to invalidate a previously signed certificate so that it can no longer be used for authentication purposes.

Typical reasons for wanting to revoke a certificate include:

* The private key associated with the certificate is compromised or stolen.

* The user of an encrypted private key forgets the password on the key.

* You want to terminate a VPN user's access.

Example:

As an example, we will revoke the client2 certificate, which we generated above in the "key generation" section of this application note.

First open up a command prompt window and cd to the easy-rsa directory as you did in the "key generation" section above.

On Windows, type:

```
    vars
    revoke-full client2
```

You should see output similar to this:

```
    Using configuration from C:\Program Files\OpenVPN\easy-rsa\openssl.cnf
    DEBUG[load_index]: unique_subject = "yes"
    Revoking Certificate 04.
    Data Base Updated
    Using configuration from C:\Program Files\OpenVPN\easy-rsa\openssl.cnf
    DEBUG[load_index]: unique_subject = "yes"
    client2.crt: /C=UK/ST=West-Yorkshire/O=Digi-
UK/CN=client2/emailAddress=uksupport@digi.com
    error 23 at 0 depth lookup:certificate revoked
```

Note the "error 23" in the last line. That is what you want to see, as it indicates that a certificate verification of the revoked certificate failed.

The revoke-full script will generate a CRL (certificate revocation list) file called crl.pem in the keys subdirectory. This file should be copied onto the server in the config directory and replaced every time a certificate is revoked.

Now all connecting clients will have their client certificates verified against the CRL, and any positive match will result in the connection being dropped.

7 FIRMWARE VERSIONS

7.1 Digi TransPort WR

```
Digi TransPort WR21-U22B-DE1-XX Ser#:237416
Software Build Ver5.2.19.6.  Aug 23 2017 11:05:52  WW
ARM Bios Ver 7.61u v43 454MHz B987-M995-F80-O8140,0 MAC:00042d039f68
Async Driver              Revision: 1.19  Int clk
Ethernet Port Isolate Driver Revision: 1.11
Firewall                  Revision: 1.0
EventEdit                 Revision: 1.0
Timer Module              Revision: 1.1
(B)USBHOST                Revision: 1.0
L2TP                      Revision: 1.10
PPTP                      Revision: 1.00
TACPLUS                   Revision: 1.00
MODBUS                    Revision: 0.00
RealPort                  Revision: 0.00
MultiTX                   Revision: 1.00
LAPB                      Revision: 1.12
X25 Layer                 Revision: 1.19
MACRO                     Revision: 1.0
PAD                       Revision: 1.4
X25 Switch                Revision: 1.7
V120                      Revision: 1.16
TPAD Interface            Revision: 1.12
GPS                       Revision: 1.0
TELITUPD                  Revision: 1.0
SCRIBATSK                 Revision: 1.0
BASTSK                    Revision: 1.0
PYTHON                    Revision: 1.0
CLOUDSMS                  Revision: 1.0
TCP (HASH mode)           Revision: 1.14
TCP Utils                 Revision: 1.13
PPP                       Revision: 5.2
WEB                       Revision: 1.5
SMTP                      Revision: 1.1
FTP Client                Revision: 1.5
FTP                       Revision: 1.5
IKE                       Revision: 1.0
PollANS                   Revision: 1.2
PPPOE                     Revision: 1.0
BRIDGE                    Revision: 1.1
MODEM CC (Huawei LTE)     Revision: 5.2
FLASH Write               Revision: 1.2
Command Interpreter       Revision: 1.38
SSLCLI                    Revision: 1.0
OSPF                      Revision: 1.0
BGP                       Revision: 1.0
QOS                       Revision: 1.0
PWRCTRL                   Revision: 1.0
RADIUS Client             Revision: 1.0
```

```
SSH Server                Revision: 1.0
SCP                       Revision: 1.0
SSH Client                Revision: 1.0
CERT                      Revision: 1.0
LowPrio                   Revision: 1.0
Tunnel                    Revision: 1.2
OVPN                      Revision: 1.2
TEMPLOG                   Revision: 1.0
QDL                       Revision: 1.0
OK
```

7.2 Windows OpenVPN Client 1

```
C:\windows\system32>openvpn --version
OpenVPN 2.2.2 Win32-MSVC++ [SSL] [LZO2] [PKCS11] built on Dec 15 2011
Originally developed by James Yonan
Copyright (C) 2002-2010 OpenVPN Technologies, Inc. <sales@openvpn.net>

 config_all.py

Compile time defines:  ENABLE_HTTP_PROXY=1, ENABLE_DEBUG=1, ENABLE_MANAGEMENT=1,
ENABLE_CLIENT_SERVER=1, ENABLE_PASSWORD_SAVE=1, ENABLE_CLIENT_ONLY=0, ENABLE_SOCKS=1,
ENABLE_FRAGMENT=1,
```

7.3 Windows OpenVPN Client 2

```
C:\Users\INGTest>openvpn --version
OpenVPN 2.2.2 Win32-MSVC++ [SSL] [LZO2] [PKCS11] built on Dec 15 2011
Originally developed by James Yonan
Copyright (C) 2002-2010 OpenVPN Technologies, Inc. <sales@openvpn.net>

 config_all.py

Compile time defines:  ENABLE_HTTP_PROXY=1, ENABLE_DEBUG=1, ENABLE_MANAGEMENT=1,
ENABLE_CLIENT_SERVER=1, ENABLE_PASSWORD_SAVE=1, ENABLE_CLIENT_ONLY=0, ENABLE_SOCKS=1,
ENABLE_FRAGMENT=1,
```

8 CONFIGURATION FILES

8.1 Digi Transport WR

```
eth 0 IPaddr "172.16.0.1"
eth 1 dhcpcli ON
lapb 0 ans OFF
lapb 0 tinact 120
lapb 1 tinact 120
lapb 3 dtemode 0
lapb 4 dtemode 0
lapb 5 dtemode 0
lapb 6 dtemode 0
ip 0 cidr ON
route 0 descr "toClient1LAN"
route 0 IPaddr "172.16.1.0"
route 0 ll_ent "OVPN"
route 1 descr "toClient2LAN"
route 1 IPaddr "172.16.2.0"
route 1 ll_ent "OVPN"
route 1 ll_add 1
def_route 0 ll_ent "PPP"
def_route 0 ll_add 1
dhcp 0 IPmin "192.168.1.100"
dhcp 0 respdelms 500
dhcp 0 mask "255.255.255.0"
dhcp 0 gateway "192.168.1.1"
dhcp 0 DNS "192.168.1.1"
sntp 0 server "time.devicecloud.com"
dyndns 0 ifent "default"
ppp 0 timeout 300
ppp 1 name "W-WAN (LTE)"
ppp 1 phonenum "*98*1#"
ppp 1 IPaddr "0.0.0.0"
ppp 1 timeout 0
ppp 1 use_modem 1
ppp 1 aodion 1
ppp 1 autoassert 1
ppp 1 r_chap OFF
ppp 3 defpak 16
ppp 4 defpak 16
web 0 prelogin_info ON
web 0 showgswiz ON
modemcc 0 info_asy_add 4
modemcc 0 init_str "+CGQREQ=1"
modemcc 0 init_str1 "+CGQMIN=1"
modemcc 0 apn "internet.t-d1.de"
modemcc 0 link_retries 10
modemcc 0 stat_retries 30
modemcc 0 sms_interval 1
modemcc 0 sms_access 1
modemcc 0 sms_concat 0
```

```
modemcc 0 init_str_2 "+CGQREQ=1"
modemcc 0 init_str1_2 "+CGQMIN=1"
modemcc 0 apn_2 "Your.APN.goes.here"
modemcc 0 link_retries_2 10
modemcc 0 stat_retries_2 30
modemcc 0 sms_access_2 1
modemcc 0 sms_concat_2 0
ana 0 l1on ON
ana 0 lapdon 0
ana 0 asyon 1
ana 0 logsize 45
cmd 0 unitid "ss%s>"
cmd 0 cmdnua "99"
cmd 0 hostname "digi.router"
cmd 0 asyled_mode 2
cmd 0 tremto 1200
cmd 0 rcihttp ON
user 0 access 0
user 1 name "username"
user 1 epassword "KD5lSVJDVVg="
user 1 access 0
user 2 access 0
user 3 access 0
user 4 access 0
user 5 access 0
user 6 access 0
user 7 access 0
user 8 access 0
user 9 access 0
local 0 transaccess 2
sslcli 0 verify 10
sslsvr 0 certfile "certserv.pem"
sslsvr 0 keyfile "privserv.pem"
ssh 0 hostkey1 "privSSH.pem"
ssh 0 nb_listen 5
ssh 0 v1 OFF
ovpn 0 descr "toClient1"
ovpn 0 IPaddr "192.168.0.1"
ovpn 0 server ON
ovpn 0 puship "172.16.0.0"
ovpn 0 pushmask "255.255.255.0"
ovpn 0 pushdns "8.8.8.8"
ovpn 0 cipher "AES-256-CBC"
ovpn 1 descr "toClient2"
ovpn 1 IPaddr "192.168.0.5"
ovpn 1 server ON
ovpn 1 puship "172.16.0.0"
ovpn 1 pushmask "255.255.255.0"
ovpn 1 pushdns "8.8.8.8"
ovpn 1 pingint 10
ovpn 1 pingto 120
ovpn 1 cipher "AES-256-CBC"
templog 0 mo_autooff ON
cloud 0 ssl ON
```

8.2 Windows OpenVPN Client 1

```
##############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
##############################################

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap

# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 10.104.1.115 1194
;remote my-server-2 1194

# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
```

```
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nobody

# Try to preserve some state across restarts.
persist-key
persist-tun

# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here.  See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets.  Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings

# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca ca.crt
cert client1.crt
key client1.key

# Verify server certificate by checking that the
# certicate has the correct key usage set.
# This is an important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the keyUsage set to
#   digitalSignature, keyEncipherment
# and the extendedKeyUsage to
#   serverAuth
# EasyRSA can do this for you.
remote-cert-tls server

# If a tls-auth key is used on the server
```

```
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
# Note that 2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
cipher AES-256-CBC


# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
;comp-lzo

# Set log file verbosity.
verb 3

# This whole section has been added and is important
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120


# Silence repeating messages
;mute 20
```

8.3 Windows OpenVPN Client 2

```
#############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
#############################################

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap

# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 10.104.1.115 1194
;remote my-server-2 1194

# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
```

```
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nobody

# Try to preserve some state across restarts.
persist-key
persist-tun

# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here.  See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets.  Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings

# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca ca.crt
cert client2.crt
key client2.key

# Verify server certificate by checking that the
# certicate has the correct key usage set.
# This is an important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the keyUsage set to
#   digitalSignature, keyEncipherment
# and the extendedKeyUsage to
#   serverAuth
# EasyRSA can do this for you.
remote-cert-tls server

# If a tls-auth key is used on the server
```

```
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
# Note that 2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
cipher AES-256-CBC


# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
;comp-lzo

# Set log file verbosity.
verb 3

# This whole section has been added and is important
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120


# Silence repeating messages
;mute 20
```

9 APPENDIX 1

9.1 Throughput test results

The following testing was done using the same configuration and topology detailed in this application note. The router, server and user laptops were all connected via Ethernet only. Throughput was measured with the iperf throughput testing application.

Routed connection on Ethernet between laptop and server, no VPN active.

Test duration: 30 seconds

Data transferred: 159Mb

Throughput: 44.6 Mbit/sec

1 OpenVPN client connected via Ethernet.

Test duration: 30 seconds

Data transferred: 37Mb

Throughput: 9.9 Mbit/sec

2 OpenVPN clients connected via Ethernet.

Test duration: 30 seconds

Total Data transferred: 37Mb

Client 1 throughput: 5.08 Mbit/sec

Client 2 throughput: 4.77 Mbit/sec

9.2 OpenVPN vs IPsec

There are many differences between OpenVPN and IPsec, it is down to the network administrator to make the decision about which VPN solution to use.

OpenVPN is generally easier for the end user to work with and simpler to configure than IPsec, due to the client software being installed on the user's PC or laptop. Also, the network administrator can pre-configure OpenVPN client configuration files and create certificates ready for copying across to the user's PC or laptop.

IPsec functions are built into Windows, Linux & UNIX platforms as standard, so no extra client software is required to be installed, but a knowledge of configuring IPsec is generally required as it is more complex to set up.

However, the throughput of OpenVPN is much lower than that of IPsec and as such it may not be suitable for large scale deployment. If multiple concurrent users require VPN access to a corporate LAN, then IPsec will probably be the better option.

There is plenty of information available on the internet regarding this subject, just browse to your favourite search engine and type "OpenVPN Vs IPsec".