



# **Application Note 47**

---

Configuring a Windows OpenVPN server and a  
TransPort WR as an OpenVPN client

## Contents

1	Introduction .....	4
1.1	Outline .....	4
1.2	Assumptions .....	5
1.3	Corrections .....	5
1.4	Version .....	5
2	OpenVPN & Easy-RSA setup.....	6
2.1	Download the OpenVPN installation package and install the software.....	6
2.2	Setting up Certificate Authority (CA) and generating certificates and keys .....	10
2.3	Generate the master Certificate Authority (CA) certificate & key.....	11
2.3.1	Generate certificate & key for server .....	14
2.3.2	Generate certificates & keys for the client .....	15
2.3.3	Generate Diffie Hellman parameters.....	16
2.3.4	Key Files .....	17
3	Windows OpenVPN server configuration .....	18
3.1	Install the OpenVPN software .....	18
3.2	Install the SSL certificates.....	18
3.3	Configure the OpenVPN Server (server.ovpn) .....	19
3.4	Start the OpenVPN Server .....	25
4	TransPort WR configuration .....	27
4.1	WAN Interface configuration .....	27
4.2	LAN Interface configuration.....	28
4.3	Transfer Certificates and Key files .....	29
4.4	SSL Certificates configuration .....	30
4.5	OpenVPN Client mode configuration .....	31
5	Test OpenVPN Connection.....	33
5.1	OpenVPN Connection Status.....	33
5.2	Routing Table .....	35

5.3	Check the traffic on the OpenVPN Connection .....	35
6	Revoking a certificate.....	<b>Errore. Il segnalibro non è definito.</b>
7	Firmware versions.....	37
7.1	Digi TransPort WR .....	37
7.2	Windows OpenVPN Server .....	38
8	Configuration Files .....	39
8.1	Digi Transport WR .....	39
8.2	Windows OpenVPN Server .....	41
9	Appendix: OpenVPN Vs IPsec.....	47

# 1 INTRODUCTION

## 1.1 Outline

This document describes how to configure a Windows OpenVPN server and a TransPort WR router as an OpenVPN client.

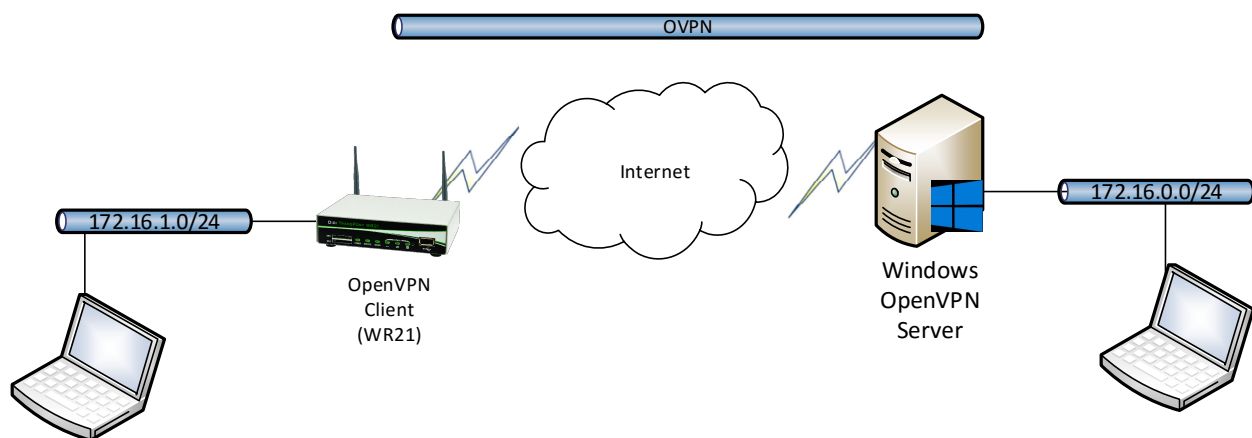
OpenVPN can be used for securely connecting the WR router to a central office network for access to services on the LAN side of the OpenVPN server, such as corporate messaging services, file servers and print servers for example.

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 following scenario will be used:



OpenVPN is certificate based, so there will be certificates on the two peers.

A PC will be needed that can be used to install the OpenVPN Easy-RSA certificate authority and create & sign the certificates.

## 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. It also assumes a basic ability to access and navigate a Digi Transport router and configure it with basic routing functions.

This application note applies to:

**Model:** Digi Transport WR21

**Other Compatible Models:** All Digi WR Transpost models

**Firmware versions:** 5.077 and later

**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.

**Software required:** OpenVPN 2.3.18, Windows Server 2016

**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@digicom.com](mailto:tech.support@digicom.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	Overall update. Testing with OpenVPN 2.3 and windows server 2016. New screenshots, new tests, added test and references to other OVPN docs. WR configuration rewritten. Layout fixes.

## 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 this example, a Windows Server 2016 is used (and this will be used also as the Open VPN server).

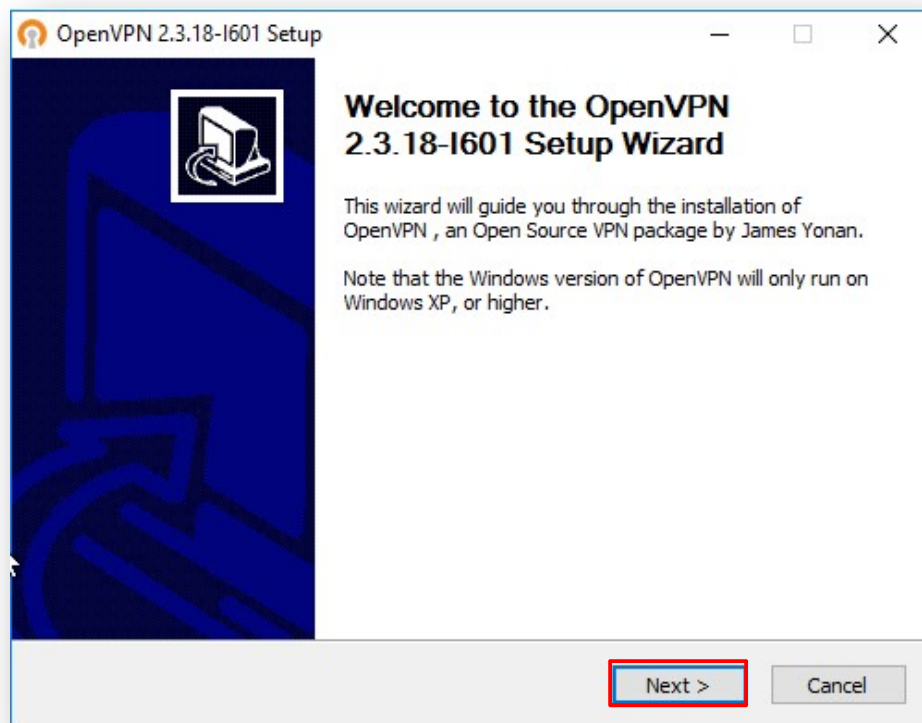
In order to download the installer for OpenVPN, go to:

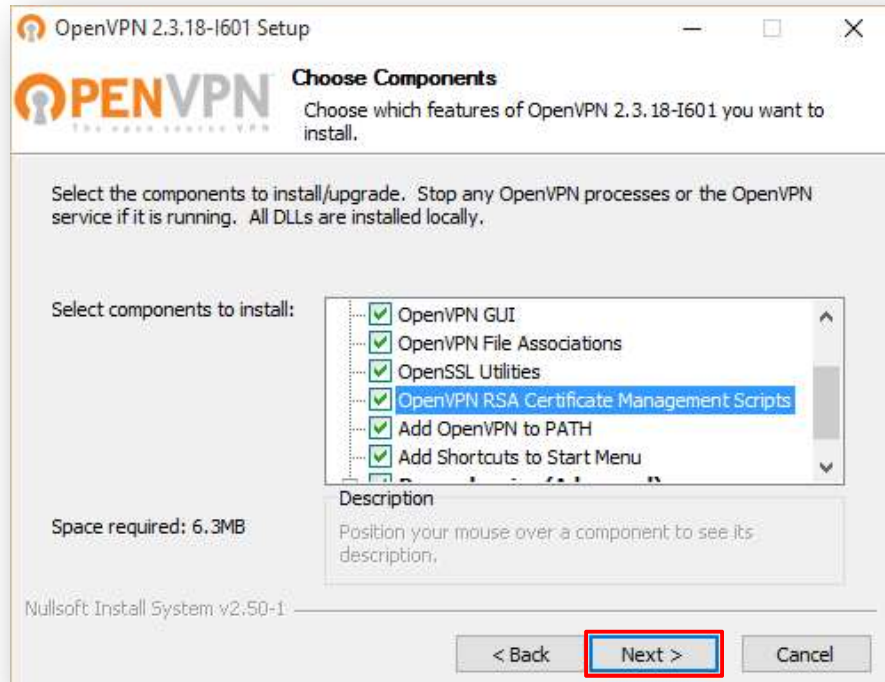
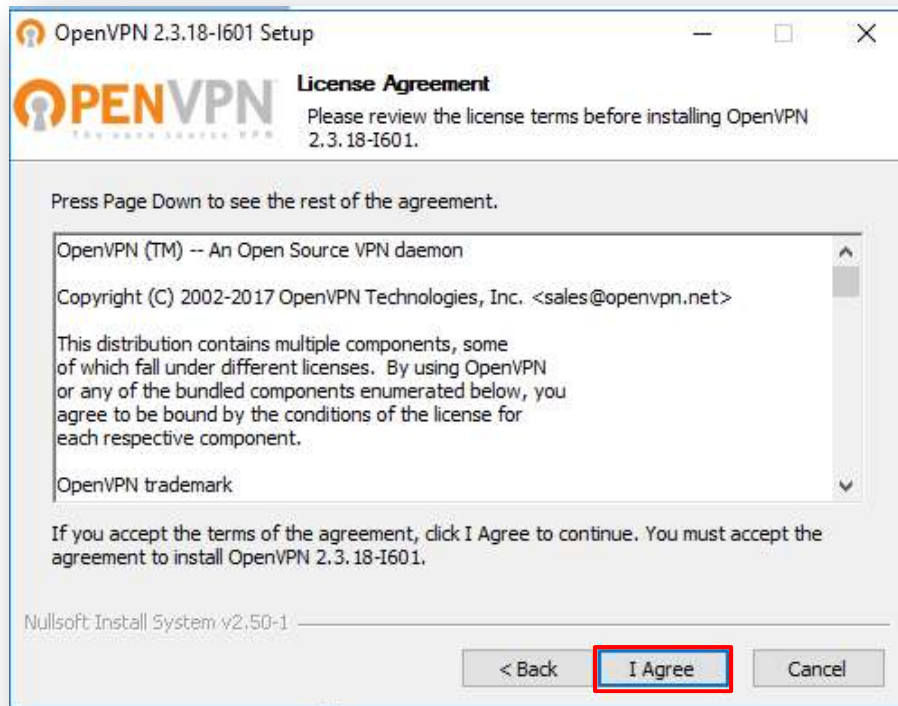
<https://openvpn.net/index.php/open-source/downloads.html>

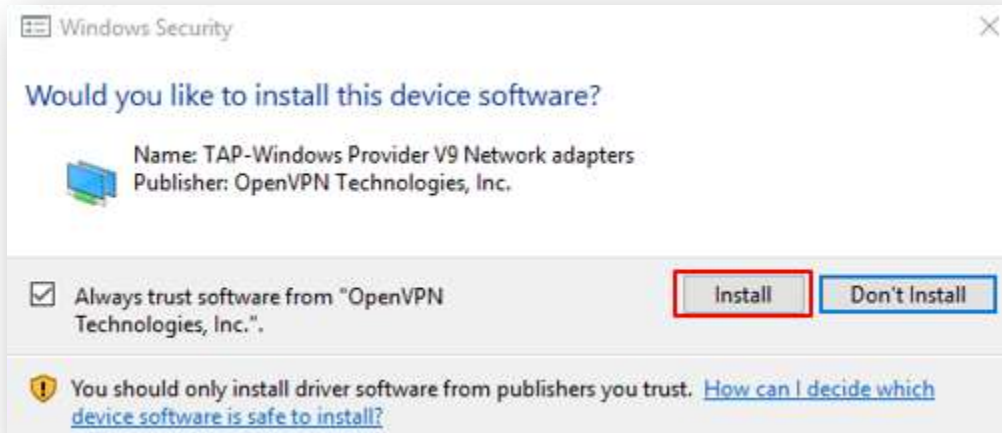
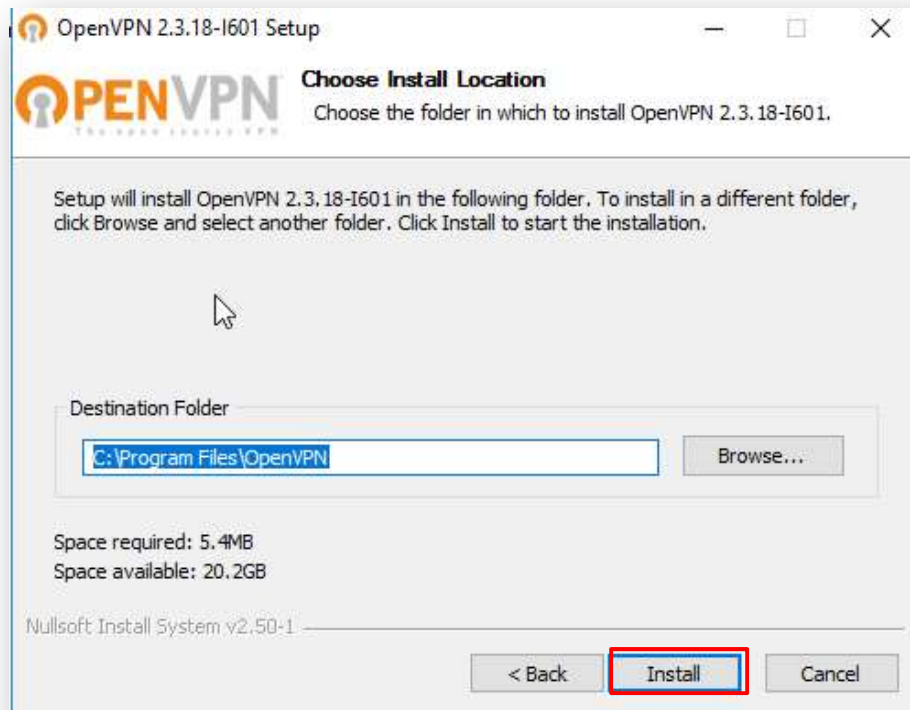
For this example, OpenVPN 2.3.18 version has been used:



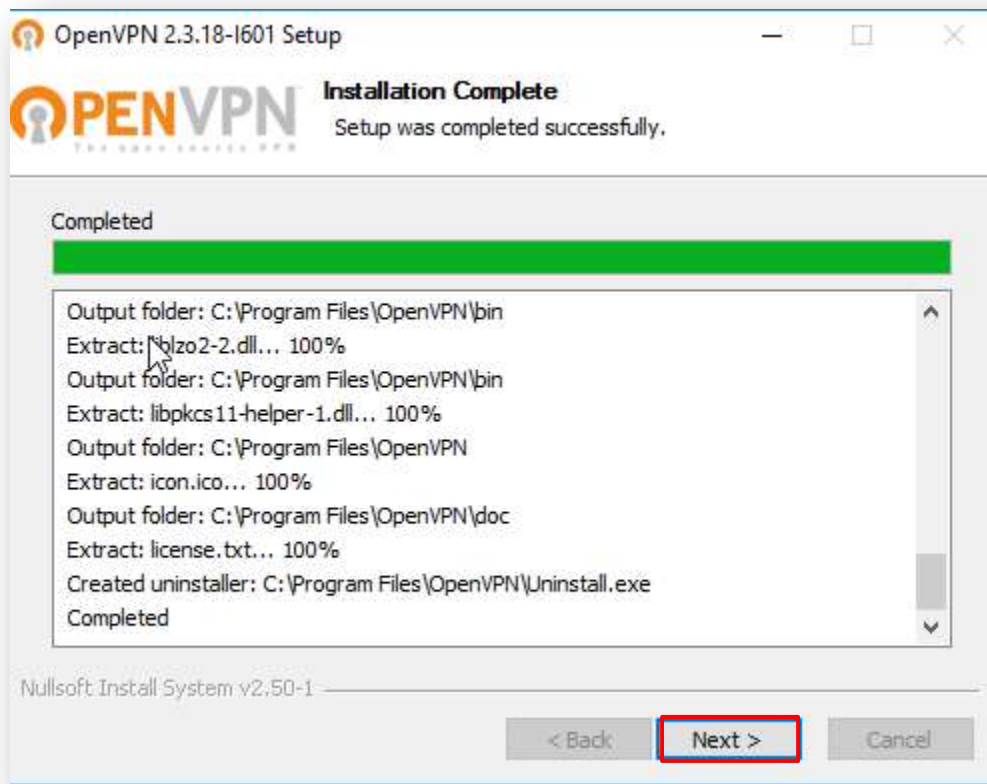
Run the installer and follow the instructions:











The installation process is completed at this point.

## 2.2 Setting up Certificate Authority (CA) and generating certificates and keys

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.

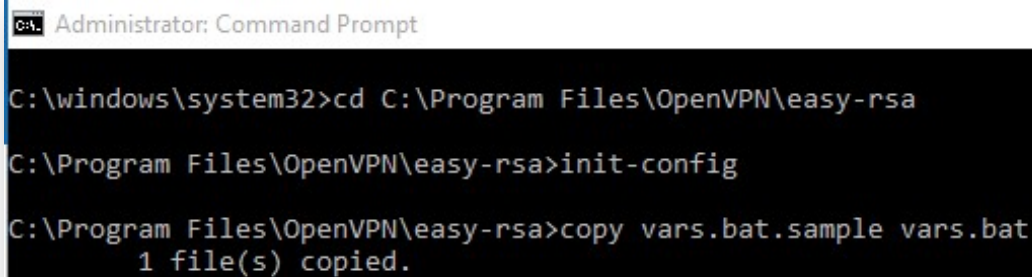
For PKI management, we will use easy-rsa, included in the OpenVPN installation.

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):

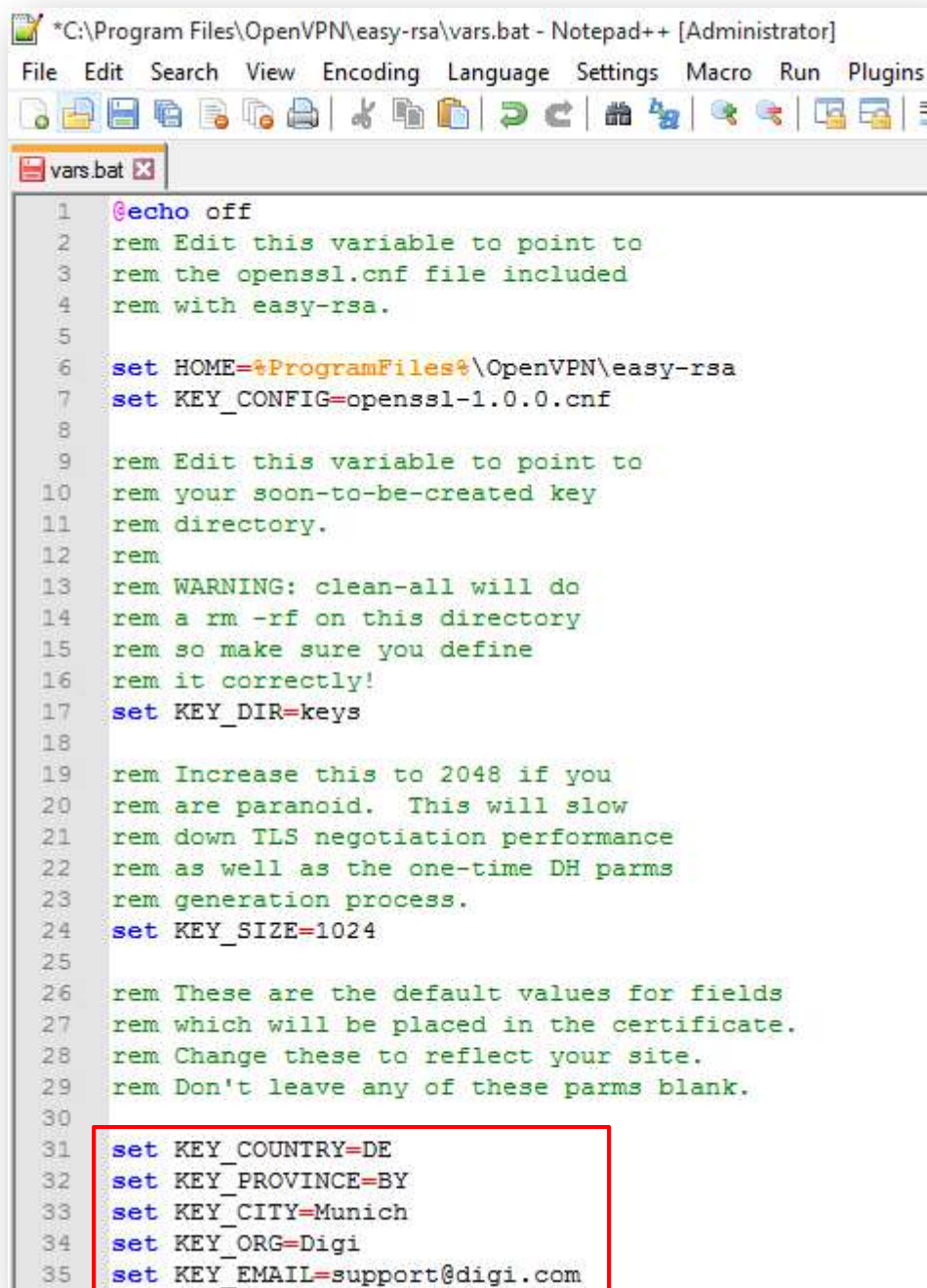
```
init-config
```

The output will be like the following:



```
Administrator: Command Prompt
C:\windows\system32>cd C:\Program Files\OpenVPN\easy-rsa
C:\Program Files\OpenVPN\easy-rsa>init-config
C:\Program Files\OpenVPN\easy-rsa>copy vars.bat.sample vars.bat
1 file(s) copied.
```

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 parameters blank:



```
*C:\Program Files\OpenVPN\easy-rsa\vars.bat - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins

vars.bat x
1 @echo off
2 rem Edit this variable to point to
3 rem the openssl.cnf file included
4 rem with easy-rsa.
5
6 set HOME=%ProgramFiles%\OpenVPN\easy-rsa
7 set KEY_CONFIG=openssl-1.0.0.cnf
8
9 rem Edit this variable to point to
10 rem your soon-to-be-created key
11 rem directory.
12 rem
13 rem WARNING: clean-all will do
14 rem a rm -rf on this directory
15 rem so make sure you define
16 rem it correctly!
17 set KEY_DIR=keys
18
19 rem Increase this to 2048 if you
20 rem are paranoid. This will slow
21 rem down TLS negotiation performance
22 rem as well as the one-time DH parms
23 rem generation process.
24 set KEY_SIZE=1024
25
26 rem These are the default values for fields
27 rem which will be placed in the certificate.
28 rem Change these to reflect your site.
29 rem Don't leave any of these parms blank.
30
31 set KEY_COUNTRY=DE
32 set KEY_PROVINCE=BY
33 set KEY_CITY=Munich
34 set KEY_ORG=Digi
35 set KEY_EMAIL=support@digi.com
```

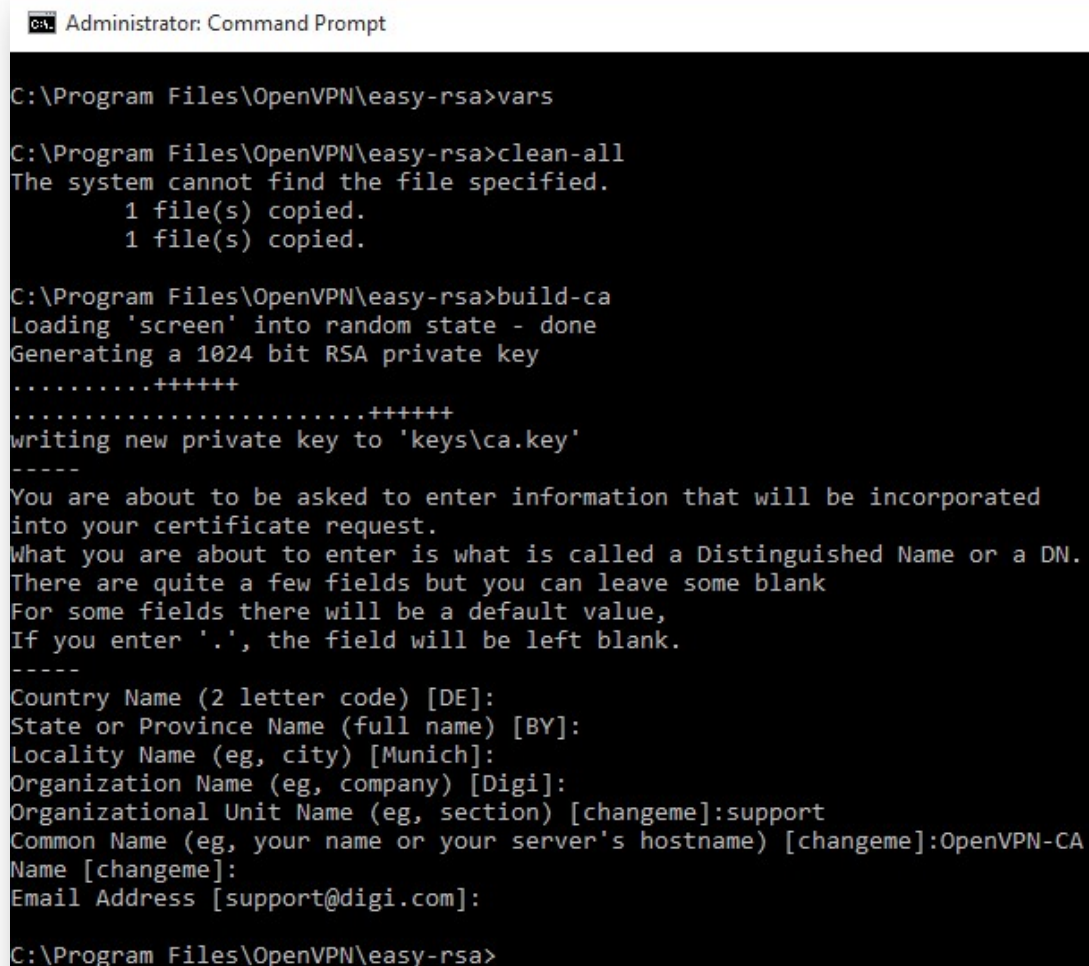
Save and close it.

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:



```
Administrator: Command Prompt

C:\Program Files\OpenVPN\easy-rsa>vars

C:\Program Files\OpenVPN\easy-rsa>clean-all
The system cannot find the file specified.
      1 file(s) copied.
      1 file(s) copied.

C:\Program Files\OpenVPN\easy-rsa>build-ca
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'keys\ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [DE]:
State or Province Name (full name) [BY]:
Locality Name (eg, city) [Munich]:
Organization Name (eg, company) [Digi]:
Organizational Unit Name (eg, section) [changeme]:support
Common Name (eg, your name or your server's hostname) [changeme]:OpenVPN-CA
Name [changeme]:
Email Address [support@digicom]:

C:\Program Files\OpenVPN\easy-rsa>
```

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]".

```
Administrator: Command Prompt
C:\Program Files\OpenVPN\easy-rsa>build-key-server server
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.....+++++
.+++++
writing new private key to 'keys\server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [DE]:
State or Province Name (full name) [BY]:
Locality Name (eg, city) [Munich]:
Organization Name (eg, company) [Digi]:
Organizational Unit Name (eg, section) [changeme]:support
Common Name (eg, your name or your server's hostname) [changeme]:server
Name [changeme]:
Email Address [support@digicom]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from openssl-1.0.0.cnf
Loading 'screen' into random state - done
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName             :PRINTABLE:'DE'
stateOrProvinceName     :PRINTABLE:'BY'
localityName            :PRINTABLE:'Munich'
organizationName        :PRINTABLE:'Digi'
organizationalUnitName  :PRINTABLE:'support'
commonName              :PRINTABLE:'server'
name                   :PRINTABLE:'changeme'
emailAddress            :IA5STRING:'support@digicom'
Certificate is to be certified until Jul 17 10:26:39 2027 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]:y
Write out database with 1 new entries
Data Base Updated

C:\Program Files\OpenVPN\easy-rsa>
```



## 2.2.3 Generate certificates & keys for the client

Generating client certificates is very similar to the previous step.

```
build-key client1
```

```
Administrator: Command Prompt
C:\Program Files\OpenVPN\easy-rsa>build-key-client client1
'build-key-client' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\OpenVPN\easy-rsa>build-key client1
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'keys\client1.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [DE]:
State or Province Name (full name) [BY]:
Locality Name (eg, city) [Munich]:
Organization Name (eg, company) [Digi]:
Organizational Unit Name (eg, section) [changeme]:
Common Name (eg, your name or your server's hostname) [changeme]:client1
Name [changeme]:
Email Address [support@digicom]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from openssl-1.0.0.cnf
Loading 'screen' into random state - done
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName             :PRINTABLE:'DE'
stateOrProvinceName     :PRINTABLE:'BY'
localityName            :PRINTABLE:'Munich'
organizationName        :PRINTABLE:'Digi'
organizationalUnitName  :PRINTABLE:'changeme'
commonName              :PRINTABLE:'client1'
name                   :PRINTABLE:'changeme'
emailAddress            :IASSTRING:'support@digicom'
Certificate is to be certified until Jul 17 10:28:35 2027 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]:y
Write out database with 1 new entries
Data Base Updated

C:\Program Files\OpenVPN\easy-rsa>
```

Remember that if you create Certificates and Keys for more than one client, for each client, make sure to type the appropriate Common Name when prompted, i.e. "client1", "client2", or "client3". Always use a unique common name for each client.

Diffie Hellman parameters must be generated for the OpenVPN server with the following command:

Output:

Page | 16



### 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

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

## 3 WINDOWS OPENVPN SERVER CONFIGURATION

The following steps explain the configuration that needs to be done on the Windows OpenVPN server.

### 3.1 Install the OpenVPN software

This step is only required if the OpenVPN server is a different PC to the one used to create RSA certificates earlier. In this example the same PC is used as per section above, so the software is already installed and ready to use.

Please follow same steps as section [2.1](#) if a new installation is needed on the server.

### 3.2 Install the SSL certificates

The SSL certificates that were created earlier should now be securely transferred to the **OpenVPN\config** directory from the Certificate Authority PC.

The files that should be moved are:

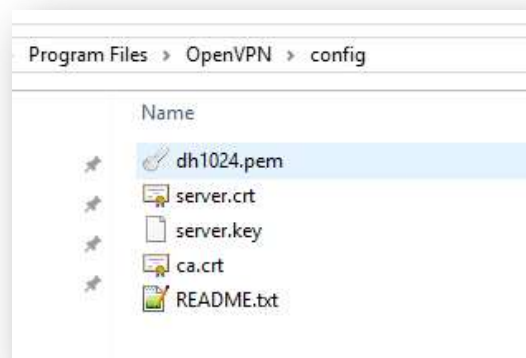
*ca.crt*

*dh1024.pem*

*server.crt*

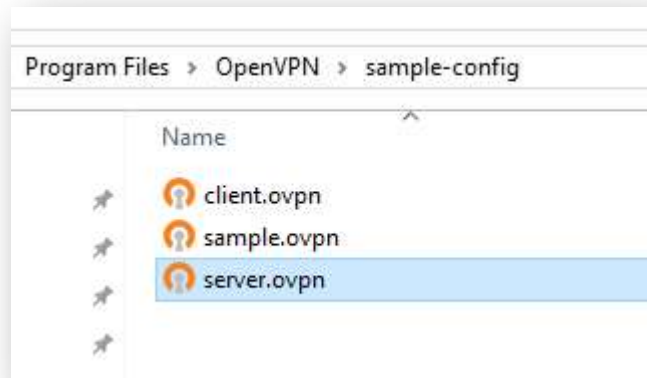
*server.key*

So in the OpenVPN\config folder there should files be like following:



### 3.3 Configure the OpenVPN Server (server.ovpn)

Open and edit the server.ovpn file from the OpenVPN\sample-config 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 OpenVPN 2.0 config file for      #
# multi-client server.                    #
#                                         #
# This file is for the server side        #
# of a many-clients <-> one-server        #
# OpenVPN configuration.                  #
#                                         #
# OpenVPN also supports                   #
# single-machine <-> single-machine       #
# configurations (See the Examples page   #
# on the web site for more info).         #
#                                         #
# This config should work on Windows     #
# or Linux/BSD systems. Remember on      #
# Windows to quote pathnames and use     #
# double backslashes, e.g.:              #
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #
#                                         #
# Comments are preceded with '#' or ';'   #
#####

# Which local IP address should OpenVPN
# listen on? (optional)
local 10.104.1.126

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
```

```

# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194

# TCP or UDP server?
;proto tcp
proto udp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# 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 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert server.crt
key server.key # This file should be kept secret

# Diffie hellman parameters.

```

```

# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh dh1024.pem

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.0.0.0 255.255.255.0

# Maintain a record of client <-> virtual IP address
# associations in this file. If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
ifconfig-pool-persist ipp.txt

# Configure server mode for ethernet bridging.
# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface. Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0. Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients. Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# Configure server mode for ethernet bridging
# using a DHCP-proxy, where clients talk
# to the OpenVPN server-side DHCP server
# to receive their IP address allocation
# and DNS server addresses. You must first use
# your OS's bridging capability to bridge the TAP
# interface with the ethernet NIC interface.
# Note: this mode only works on clients (such as
# Windows), where the client-side TAP adapter is
# bound to a DHCP client.
;server-bridge

# Push routes to the client to allow it
# to reach other private subnets behind
# the server. Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.

```

```

push "route 10.0.0.0 255.255.255.0" # This is the DHCP pool range
push "route 172.16.0.0 255.255.255.0" # This is the LAN subnet
# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN. This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2

# Suppose that you want to enable different
# firewall access policies for different groups
# of clients. There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients. See man
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
;push "redirect-gateway def1 bypass-dhcp"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS

```

```

# or WINS server addresses. CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by.opendns.com.
push "dhcp-option DNS 172.16.0.1" # This is the LAN connected DNS server
push "dhcp-option DNS 8.8.8.8" # This is an external public DNS server
# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
;client-to-client

# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names. This is recommended
# only for testing purposes. For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
;duplicate-cn

# 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

# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC # Blowfish (default)
;cipher AES-128-CBC # AES

```

```

cipher AES-256-CBC # AES 256
;cipher DES-EDE3-CBC # Triple-DES

# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
;comp-lzo # OpenVPN LZO compression is not supported on TransPort routers

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nobody

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun

# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it. Use one
# or the other (but not both).
;log openvpn.log
;log-append openvpn.log
# Set the appropriate 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
verb 4

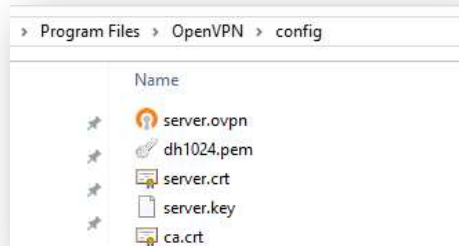
# Silence repeating messages. At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20

```



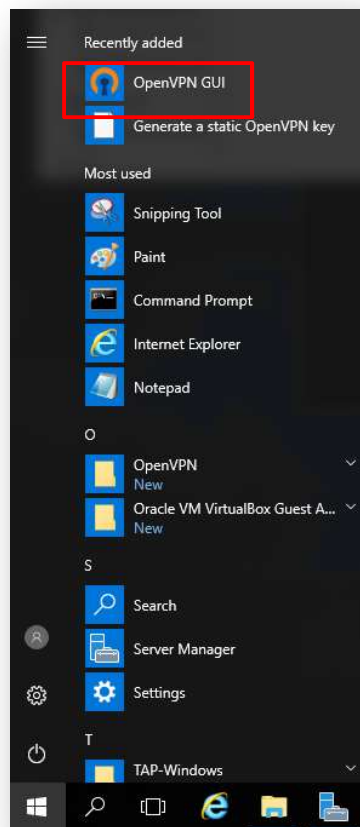
The configuration of the server is now ready for use.


In order for it to be used read from the OpenVPN software, save this file to the **OpenVPN\config** directory:



### 3.4 Start the OpenVPN Server

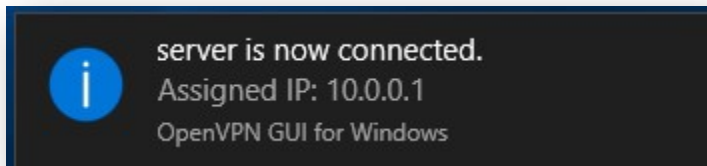
Run the OpenVPN software from the Start menu or from the desktop shortcut:



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



Double click the icon, when the OpenVPN server has successfully started, the icon will turn green and a notification of the assigned IP address will be shown:



This server will now wait for inbound OpenVPN connections.

## 4 TRANSPORT WR CONFIGURATION

### 4.1 WAN Interface configuration

In this example the Client has the Mobile interface as the WAN interface and it is configured as follows:

#### CONFIGURATION - NETWORK > INTERFACES > MOBILE

Configuration - Network > Interfaces > Mobile

▼ Interfaces

- ▶ Ethernet
- ▼ Mobile

Select a SIM to configure from the list below

Settings on this page apply to the selected SIM

SIM: 1 (PPP 1) ▼

IMSI: 262010050453499

▼ Mobile Settings

Select the service plan and connection settings used in connecting to the mobile network.

**Mobile Service Provider Settings**

Service Plan / APN: internet.t-d1.de

☐ Use backup APN Retry the main APN after 0 minutes

SIM PIN: (Optional)

Confirm SIM PIN:

Username: (Optional)

Password: (Optional)

Confirm Password:

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.

## 4.2 LAN Interface configuration

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

**CONFIGURATION - NETWORK > INTERFACES > ETHERNET > ETH 0**

Configuration - Network > Interfaces > Ethernet > ETH 0

▼ Interfaces

▼ Ethernet

▼ ETH 0

Description:

☐ Get an IP address automatically using DHCP

☒ Use the following settings

IP Address:

Mask:

Gateway:

DNS Server:

Secondary DNS Server:

Changes to these parameters may affect your browser connection

► Advanced

► QoS

► VRRP

Where:

Parameter	Setting	Description
IP Address	172.16.1.1	Enter the IP address of the LAN interface for the router
Mask	255.255.255.0	Enter the subnet mask

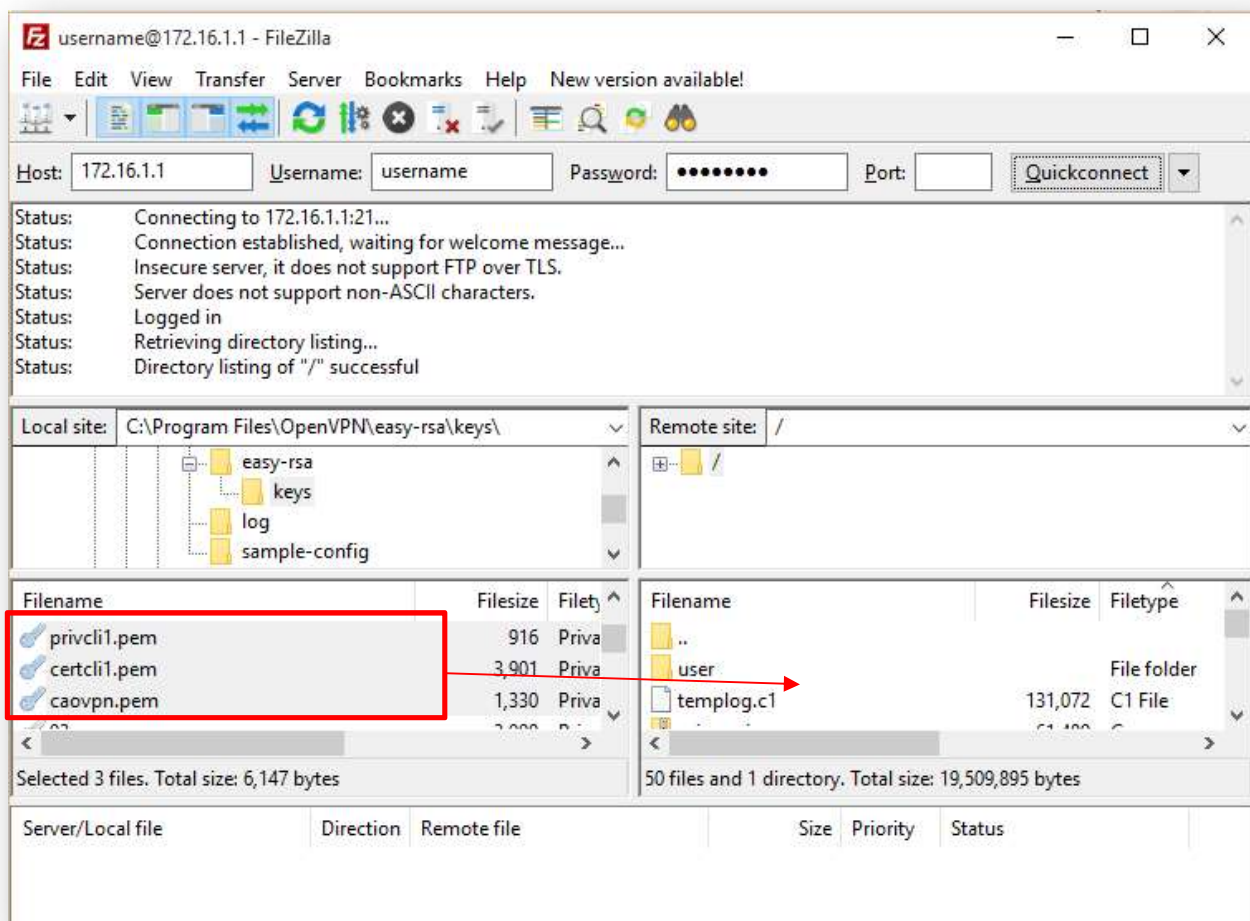
### 4.3 Transfer Certificates and Key files

Before to transfer the Certificates and Key files on the client, they must be renamed as follows as .pem files:

Filename	Purpose	New FileName
ca.crt	Root CA certificate	<b>caovpn.pem</b>
client1.crt	Client1 Certificate	<b>certcli1.pem</b>
client1.key	Client1 Key	<b>privcli1.pem</b>

Once done that, the files can be transferred to the Client 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.



## 4.4 SSL Certificates configuration

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

### CONFIGURATION – NETWORK > SSL

Configuration - Network > SSL

► Interfaces  
► DHCP Server  
► Network Services  
► DNS Servers  
► Dynamic DNS  
► IP Routing/Forwarding  
► Virtual Private Networking (VPN)  
▼ SSL

SSL Clients

SSL Client	Client Certificate Filename	Client Private Key Filename	Allow Insecure Ciphers	Cipher List	Apply to Destination IP Address	Verify Server Certificate	Reject Self-Signed Certificates
0	certcli1.pem	privcli1.pem	<input checked="" type="checkbox"/>			Also verify date	<input checked="" type="checkbox"/>
1			<input checked="" type="checkbox"/>			No	<input type="checkbox"/>
2			<input checked="" type="checkbox"/>			No	<input type="checkbox"/>
3			<input checked="" type="checkbox"/>			No	<input type="checkbox"/>
4			<input checked="" type="checkbox"/>			No	<input type="checkbox"/>
5			<input checked="" type="checkbox"/>			No	<input type="checkbox"/>

SSL Server

Server Certificate Filename	Server Private Key Filename	SSL Version	Allow Insecure Ciphers	Cipher List	Verify Certificate	Certificate Required	Reject Self-Signed Certificates
cert01.pem	privrsa.pem	TLv1.2 only	<input checked="" type="checkbox"/>		No	<input type="checkbox"/>	<input type="checkbox"/>

Apply

Where:

Parameter	Setting	Description
Client Certificate Filename	certcli1.pem	The name of the required certificate file is selected from those available on the router's filing system from this drop-down list. In this example this the one just transferred to the router.
Client Private Key Filename	privcli1.pem	The name of the file that contains the private key that matches the public key stored in the above parameter, is selected from this drop-down list. In this example this the one just transferred to the router.

## 4.5 OpenVPN Client mode configuration

An OpenVPN interface will be configured on the TransPort router that acts as OpenVPN client:

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

The screenshot shows the configuration page for 'OpenVPN 0 - toWindowsServer'. The 'Description' field is set to 'toWindowsServer'. Under the 'Use' section, the 'IP address' field is empty, and the 'Port' is set to '1194'. The 'Protocol' is set to 'UDP'. The 'Keepalive TX Interval' is '10' seconds, and the 'Keepalive RX Timeout' is '120' seconds. The 'Cipher' is set to 'AES-256-CBC' and the 'Digest' is set to 'SHA1'. For 'Route via', the 'Routing table' option is selected. The 'Source IP address' is set to 'From outgoing interface'. The 'Client Mode' radio button is selected. The 'Connect to OpenVPN server' field is set to '10.104.1.126'. Four checkboxes are checked: 'Automatically connect interface', 'Obtain IP address from the OpenVPN server', 'Obtain routes from the OpenVPN server', and 'Obtain DNS server IP address from the OpenVPN server'. The 'Server Mode' radio button is unselected. There are also options to 'Disconnect the tunnel if no IP traffic has been received for' (0 hrs 0 mins 0 secs), 'Enable NAT on this interface', and 'Enable Firewall on this interface'. An 'Advanced' section is visible at the bottom.

▼ OpenVPN

▼ OpenVPN 0 - toWindowsServer

Description: toWindowsServer

Use

IP address: Port: 1194

Protocol: UDP

Keepalive TX Interval: 10 seconds

Keepalive RX Timeout: 120 seconds

Cipher: AES-256-CBC

Digest: SHA1

Route via: ☒ Routing table

☐ Interface Auto 0

Source IP address: ☒ From outgoing interface

☐ Interface Auto 0

☒ Client Mode

Connect to OpenVPN server: 10.104.1.126

☒ Automatically connect interface

☒ Obtain IP address from the OpenVPN server

☒ Obtain routes from the OpenVPN server

☒ Obtain DNS server IP address from the OpenVPN server

☐ Server Mode

☐ Disconnect the tunnel if no IP traffic has been received for 0 hrs 0 mins 0 secs

☐ Enable NAT on this interface

☐ Enable Firewall on this interface

► Advanced

Where:

Parameter	Setting	Description
Description	toWindowsServer	Friendly name for this interface
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 with regards to protocol choice in the previous section
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
Client Mode	Selected	Use Client mode
Connect to OpenVPN server	10.104.1.126	Public IP address of OpenVPN server
Automatically connect interface	✓	Connects to the OpenVPN server automatically, always on mode.
Obtain IP address from the OpenVPN server	✓	This interface will obtain an IP address from the OpenVPN server
Obtain routes from the OpenVPN server	✓	Routing information will be obtained from the OpenVPN server
Obtain DNS Server IP address from the OpenVPN server	✓	DNS Server information will be obtained from the OpenVPN server



## 5 TEST OPENVPN CONNECTION

### 5.1 OpenVPN Connection Status

To check the OpenVPN connection status on the client, browse to:

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

▼ OpenVPN  
▼ OVPN 0

[Raise Link](#) [Drop Link](#)

Name: toWindowsServer

Uptime: 0 Hrs 6 Mins 47 Seconds

Interface IP address	10.0.0.6
Pulled Route #1	172.16.0.0/24
Pulled Route #1	10.0.0.0/24
Pulled Route #1	10.0.0.1/32
Pulled DNS server #1>	172.16.0.1
Pulled DNS server #1>	8.8.8.8
Link socket local IP	10.104.1.115
Link socket remote IP	10.104.1.126

---

Bytes Received: 94530	Bytes Sent: 147212
Packets Received: 5388	Packets Sent: 6275
Pings Received: 5316	Pings Sent: 5255
Ping Timeouts: 4	Key Renegotiations: 10
Packet Replays Detected: 3	

[Refresh](#)

This will show if the connection is active and all the network settings pushed by the server, as well as traffic statistics details.

On Server side, right-click on the icon and select “show status”, logs about client connection will be shown:

```
Mon Oct 02 03:26:44 2017 us=744371 10.104.1.115:49299 TLS: Initial packet from
[AF_INET]10.104.1.115:49299, sid=838bcded 5d8ce401
Mon Oct 02 03:26:45 2017 us=290382 10.104.1.115:49299 VERIFY OK: depth=1, C=DE,
ST=BY, L=Munich, O=Digi, OU=support, CN=OpenVPN-CA, name=changeme,
emailAddress=support@digi.com
Mon Oct 02 03:26:45 2017 us=290382 10.104.1.115:49299 VERIFY OK: depth=0, C=DE,
ST=BY, L=Munich, O=Digi, OU=support, CN=client1, name=changeme,
emailAddress=support@digi.com
...
Mon Oct 02 03:26:45 2017 us=290382 10.104.1.115:49299 Data Channel Encrypt: Cipher
'AES-256-CBC' initialized with 256 bit key
Mon Oct 02 03:26:45 2017 us=290382 10.104.1.115:49299 Data Channel Encrypt: Using 160
bit message hash 'SHA1' for HMAC authentication
Mon Oct 02 03:26:45 2017 us=290382 10.104.1.115:49299 Data Channel Decrypt: Cipher
'AES-256-CBC' initialized with 256 bit key
Mon Oct 02 03:26:45 2017 us=290382 10.104.1.115:49299 Data Channel Decrypt: Using 160
bit message hash 'SHA1' for HMAC authentication
Mon Oct 02 03:26:45 2017 us=290382 10.104.1.115:49299 Control Channel: TLSv1.2,
cipher TLSv1/SSLv3 DHE-RSA-AES256-GCM-SHA384, 1024 bit RSA
Mon Oct 02 03:26:45 2017 us=290382 10.104.1.115:49299 [client1] Peer Connection
Initiated with [AF_INET]10.104.1.115:49299
Mon Oct 02 03:26:45 2017 us=290382 client1/10.104.1.115:49299 MULTI_sva: pool
returned IPv4=10.0.0.6, IPv6=(Not enabled)
Mon Oct 02 03:26:45 2017 us=290382 client1/10.104.1.115:49299 MULTI: Learn: 10.0.0.6
-> client1/10.104.1.115:49299
Mon Oct 02 03:26:45 2017 us=290382 client1/10.104.1.115:49299 MULTI: primary virtual
IP for client1/10.104.1.115:49299: 10.0.0.6
Mon Oct 02 03:26:45 2017 us=320011 client1/10.104.1.115:49299 PUSH: Received control
message: 'PUSH_REQUEST'
Mon Oct 02 03:26:45 2017 us=320011 client1/10.104.1.115:49299 send_push_reply():
safe_cap=940
Mon Oct 02 03:26:45 2017 us=320011 client1/10.104.1.115:49299 SENT CONTROL [client1]:
'PUSH_REPLY,route 172.16.0.0 255.255.255.0,route 10.0.0.0 255.255.255.0,dhcp-option
DNS 172.16.0.1,dhcp-option DNS 8.8.8.8,route 10.0.0.1,topology net30,ping 10,ping-
restart 120,ifconfig 10.0.0.6 10.0.0.5' (status=1)
```

## 5.2 Routing Table

To better check that all routing information are correct in order to have the connection working as expected, check the routing table:

### MANAGEMENT - NETWORK STATUS > IP ROUTING TABLE



Destination	Gateway	Metric	Protocol	Idx	Interface	Status
10.0.0.1/32	10.0.0.5	0	OVPN	-	OVPN 0	UP
10.0.0.4/30	10.0.0.6	1	Local	-	OVPN 0	UP
10.0.0.0/24	10.0.0.5	0	OVPN	-	OVPN 0	UP
10.104.1.0/24	10.104.1.115	1	Local	-	ETH 1	UP
172.16.0.0/24	10.0.0.5	0	OVPN	-	OVPN 0	UP

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

## 5.3 Check the traffic on the OpenVPN Connection

Ping the OpenVPN Server address from the TransPort WR:



**Administration - Execute a command**

Command:

---

Command: ping 10.0.0.1  
Command result

Pinging Addr [10.0.0.1]

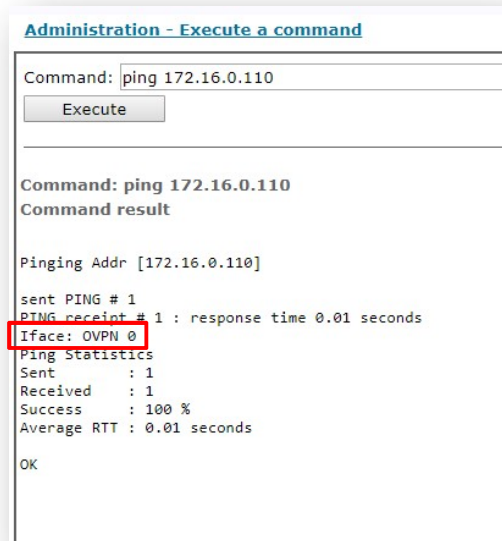
sent PING # 1  
PING receipt # 1 : response time 0.05 seconds  
Iface: OVPN 0

Ping Statistics

Sent	: 1
Received	: 1
Success	: 100 %
Average RTT	: 0.05 seconds

OK

Ping the OpenVPN Server LAN address from the TransPort WR:



Both Ping will be successful and will be sent via the OpenVPN interface OVPN0.

## 6 FIRMWARE VERSIONS

### 6.1 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-F80-08140,0 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	

## 6.2 Windows OpenVPN Server

```
C:\Users\Administrator>openvpn --version
OpenVPN 2.3.18 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO] [PKCS11] [IPv6] built on Sep
26 2017
library versions: OpenSSL 1.0.2l 25 May 2017, LZO 2.10
Windows version 6.2 (Windows 8 or greater) 64bit
Originally developed by James Yonan
Copyright (C) 2002-2017 OpenVPN Technologies, Inc. <sales@openvpn.net>
Compile time defines: enable_crypto=yes enable_crypto_ofb_cfb=yes enable_debug=yes
enable_def_auth=yes enable_dlopen=unknown enable_dlopen_self=unknown
enable_dlopen_self_static=unknown enable_fast_install=needless enable_fragment=yes
enable_http_proxy=yes enable_iproute2=no enable_libtool_lock=yes enable_lzo=yes
enable_lzo_stub=no enable_management=yes enable_multi=yes enable_multihome=yes
enable_pam_dlopen=no enable_pedantic=no enable_pf=yes enable_pkcs11=yes
enable_plugin_auth_pam=no enable_plugin_down_root=no enable_plugins=yes
enable_port_share=yes enable_selinux=no enable_server=yes enable_shared=yes
enable_shared_with_static_runtimes=yes enable_small=no enable_socks=yes
enable_ssl=yes enable_static=yes enable_strict=no enable_strict_options=no
enable_systemd=no enable_win32_dll=yes enable_x509_alt_username=no
with_crypto_library=openssl with_gnu_ld=yes with_mem_check=no
with_plugindir='${libdir}/openvpn/plugins' with_special_build= with_sysroot=no
```

## 7 CONFIGURATION FILES

### 7.1 Digi Transport WR

```
config c show
eth 0 IPaddr "172.16.1.1"
eth 0 ipanon 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
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 anon ON
ana 0 l2on OFF
ana 0 xoton OFF
ana 0 lapdon 0
ana 0 lapbon 0
ana 0 maxdata 1500
ana 0 logsize 180
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 "KD51SVJDVg="
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 certfile "certcli1.pem"
sslcli 0 keyfile "privcli1.pem"
sslcli 0 verify 10
sslsvr 0 certfile "cert01.pem"
sslsvr 0 keyfile "privrsa.pem"
ssh 0 hostkey1 "privSSH.pem"
ssh 0 nb_listen 5
ssh 0 v1 OFF
ovpn 0 descr "toWindowsServer"
ovpn 0 dest "10.104.1.126"
ovpn 0 autoup ON
ovpn 0 ipanon ON
ovpn 0 pullip ON
ovpn 0 pullroute ON
ovpn 0 pulldns ON
ovpn 0 pingint 10
ovpn 0 pingto 120
ovpn 0 cipher "AES-256-CBC"
ovpn 0 debug ON
templog 0 mo_autooff ON
cloud 0 ssl ON

Power Up Profile: 0
OK
```



## 7.2 Windows OpenVPN Server

```
#####
# Sample OpenVPN 2.0 config file for      #
# multi-client server.                    #
#                                         #
# This file is for the server side        #
# of a many-clients <-> one-server        #
# OpenVPN configuration.                  #
#                                         #
# OpenVPN also supports                   #
# single-machine <-> single-machine       #
# configurations (See the Examples page   #
# on the web site for more info).         #
#                                         #
# This config should work on Windows     #
# or Linux/BSD systems. Remember on      #
# Windows to quote pathnames and use     #
# double backslashes, e.g.:              #
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #
#                                         #
# Comments are preceded with '#' or ';'   #
#####

# Which local IP address should OpenVPN
# listen on? (optional)
local 10.104.1.126

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one. You will need to
# open up this port on your firewall.
port 1194

# TCP or UDP server?
;proto tcp
proto udp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# 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 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key). Each client
# and the server must have their own cert and
# key file. The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys. Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert server.crt
key server.key # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh dh1024.pem

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.8.0.0 255.255.255.0

# Maintain a record of client <-> virtual IP address
# associations in this file. If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
ifconfig-pool-persist ipp.txt

# Configure server mode for ethernet bridging.
```

```

# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface. Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0. Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients. Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# Configure server mode for ethernet bridging
# using a DHCP-proxy, where clients talk
# to the OpenVPN server-side DHCP server
# to receive their IP address allocation
# and DNS server addresses. You must first use
# your OS's bridging capability to bridge the TAP
# interface with the ethernet NIC interface.
# Note: this mode only works on clients (such as
# Windows), where the client-side TAP adapter is
# bound to a DHCP client.
;server-bridge

# Push routes to the client to allow it
# to reach other private subnets behind
# the server. Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
push "route 10.0.0.0 255.255.255.0" # This is the DHCP pool range
push "route 172.16.0.0 255.255.255.0" # This is the LAN subnet

# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN. This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.

```

```

# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2

# Suppose that you want to enable different
# firewall access policies for different groups
# of clients. There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients. See man
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
;push "redirect-gateway def1 bypass-dhcp"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses. CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
push "dhcp-option DNS 172.16.0.1" # This is the LAN connected DNS server
push "dhcp-option DNS 8.8.8.8" # This is an external public DNS server
# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
;client-to-client

# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names. This is recommended
# only for testing purposes. For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",

```

```

# UNCOMMENT THIS LINE OUT.
;duplicate-cn

# 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

# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC          # Blowfish (default)
;cipher AES-128-CBC     # AES
cipher AES-256-CBC    # AES 256
;cipher DES-EDE3-CBC   # Triple-DES

# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
;comp-lzo # OpenVPN LZ0 compression is not supported on TransPort routers

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nobody

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.

```

```
persist-key
persist-tun

# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it. Use one
# or the other (but not both).
;log          openvpn.log
;log-append   openvpn.log
# Set the appropriate 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
verb 4

# Silence repeating messages. At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20
```

## 8 APPENDIX: 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. 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".