
Subject: OpenVPN inside VE using TUN - Can't see server's network

Posted by [bwoo](#) on Wed, 07 May 2008 04:31:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

I've followed the OpenVZ howto regarding the tun device.

I also followed the HOWTO from OpenVPN.

I can connect between my server and my laptop. But my laptop can't seem to ping or reach any other servers other than my OpenVPN server.

This is SOOOO frustrating... Any help would be appreciated.

My OpenVPN server: 192.168.0.103

My gateway: 192.168.0.1

My client network: 10.8.0.0/24

My server and client conf's are below:

```
#####
```

```
# 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 a.b.c.d
```

```
# 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 keys/ca.crt
cert keys/server.crt
key keys/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 keys/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

# 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 192.168.10.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"

```

```
push "route 192.168.0.0 255.255.255.0"
;push "route 192.168.0.1 255.255.255.255"
;push "route 10.8.0.1 255.255.255.255"
```

```
# 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
# the TUN/TAP interface to the internet in
# order for this to work properly).
# CAVEAT: May break client's network config if
# client's local DHCP server packets get routed
# through the tunnel. Solution: make sure
# client's local DHCP server is reachable via
# a more specific route than the default route
# of 0.0.0.0/0.0.0.0.
;push "redirect-gateway"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses. CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
;push "dhcp-option DNS 10.8.0.1"
;push "dhcp-option WINS 10.8.0.1"
push "dhcp-option DNS 192.168.0.104"

# 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 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

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 3

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

#####
# 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 my-server-1 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 keys/ca.crt
cert keys/ben.crt
key keys/ben.key

# Verify server certificate by checking
# that the certificate has the nsCertType
# field set to "server". 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 nsCertType
# field set to "server". The build-key-server
# script in the easy-rsa folder will do this.
;ns-cert-type 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.
;cipher x

# 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

Silence repeating messages
;mute 20
