Pfsense and OpenVPN for new users

Author Gino Thomas
28.09.2006 thomas0@fhm.edu

14/10/2006 – Added Site-to-Site OpenVPN Sample

11/10/2006 – Added "Easy-RSA for Windows" submitted by Hernan Maslowski hernan_maslowski@hotmail.com

29/09/2006 – Removed tun0 stuff which is not needed anymore

About

This document is a step-by-step tutorial how to get pfsense and OpenVPN running. I will use many pictures and examples to make this tutorial as easy as possible. In this tutorial I used the latest snapshot and pfsense will boot from cdrom and store configuration files to floppy.

Since English is not my native language, just drop me a note if you find any mistakes. If anything is still not easy to understand let me know, too.

What you need

- Normal i386 Hardware with CDROM and floppy disk support
- At least two supported network adapters
- a place where you can burn iso images
- MS-DOS formatted floppy disk

Verify that everything is supported on FreeBSD by checking the HCL:
http://www.freebsd.org/releases/6.1R/hardware-i386.html

Downloading and burning the image

Fetch the latest version from:
http://pfsense.com/ (you want the pfSense.iso)
pfSense 1.0 RELEASE is available now

Burn the image with your favourite burning software,
Nero on windows will do the job for example. When you're done,
format the empty floppy disk with FAT.

First Boot

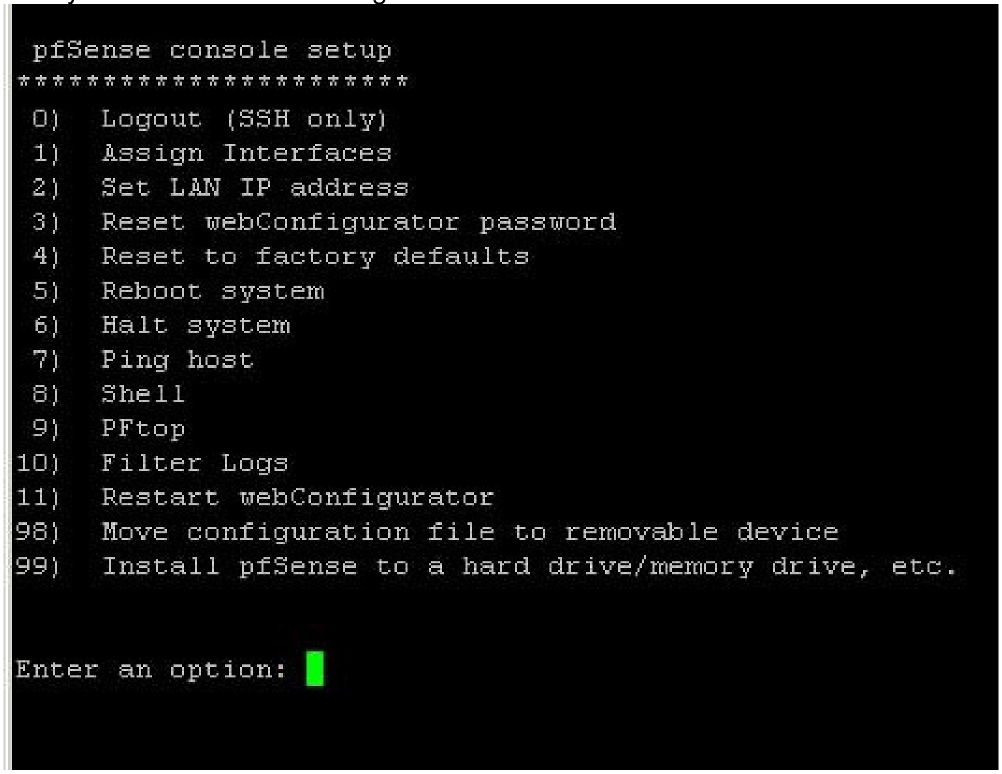
Insert the fresh burned pfsense cdrom and the FAT floppy disk in your box and configure your bios to boot from cdrom first. It is important that the floppy is inserted while booting pfsense, otherwise you won't have the option to save your configuration to disk.

After some Kernel stuff you will be asked if you want to create VLAN's, you don't want that right now so enter "n".

Next, pfsense tries to assign your network interfaces, you may use the "a" option to do an automatic lookup (you need to connect only the LAN interface to your network), but normally it's not a big deal to guess which is your LAN interface.

Enter the name of the interface shown on top of the screen to assign it as your LAN connection. After that enter the second interface name to assign it as your WAN connection, just press enter in the next step to continue. If you need more interfaces (DMZ for example) you can assign them later over the web interface.

Now you should see something like this:



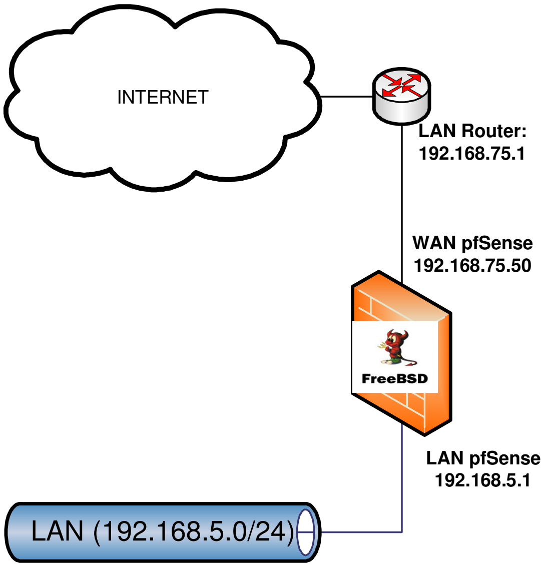
```
 pfSense console setup
***********************
 0)  Logout (SSH only)
 1)  Assign Interfaces
 2)  Set LAN IP address
 3)  Reset webConfigurator password
 4)  Reset to factory defaults
 5)  Reboot system
 6)  Halt system
 7)  Ping host
 8)  Shell
 9)  PFtop
10)  Filter Logs
11)  Restart webConfigurator
98)  Move configuration file to removable device
99)  Install pfSense to a hard drive/memory drive, etc.


Enter an option: 
```

Press "2" to enter the ip address and netmask four your LAN connection.
(In my case "192.168.5.1" and "24" as netmask, said "n" to dhcp).

Webinterface
If everything went well you should now be able to connect to the web interface of pfsense. Switch to a box on your LAN subnet and point your browser to http://ip_you_entered_for_LAN/ you should see the welcome screen of pfsense (user:admin, password:pfsense).

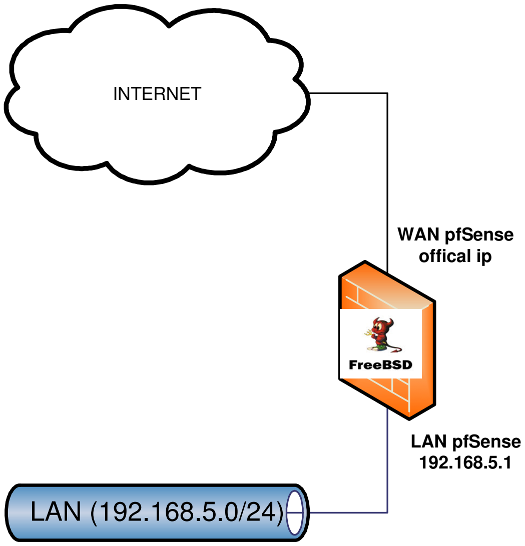
If that's not the case you most likely assigned the wrong network card as your LAN interface, repeat the "First boot" step but change the interfaces for WAN and LAN. If that still does not work, something might be wrong with your cable connection.



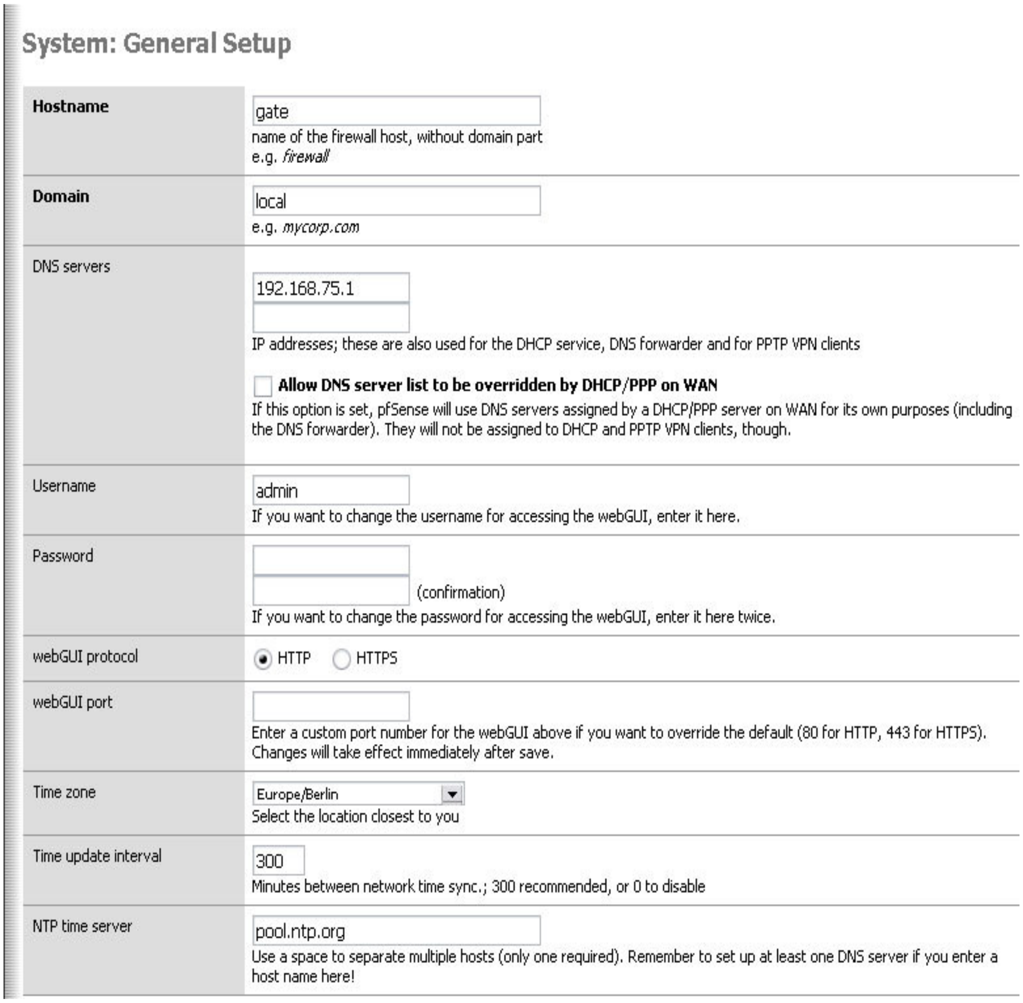
First Setup

Now we have to enter some information which depends on how your network layout is designed. In my case pfsense is a standalone box behind a 3com router which does pppoe to my isp. So my WAN interface is an internal ip (192.168.75.50) with the 3com as gateway (192.168.75.1). This is a typical situation for office networks, you naturally may have other subnets.

The second typical situation is, that there is no router for pppoe, instead pfsense is doing that job.



Whatever your layout is you have to modify the fields to your need, I will try to explain them in detail so you can decide what’s right for you.
Back to the web Interface, click on “System→ General Setup” and you will see something similar to this:



System: General Setup

Hostname	gate name of the firewall host, without domain part e.g. *firewall*
Domain	local e.g. *mycorp.com*
DNS servers	192.168.75.1 IP addresses; these are also used for the DHCP service, DNS forwarder and for PPTP VPN clients ☐ **Allow DNS server list to be overridden by DHCP/PPP on WAN** If this option is set, pfSense will use DNS servers assigned by a DHCP/PPP server on WAN for its own purposes (including the DNS forwarder). They will not be assigned to DHCP and PPTP VPN clients, though.
Username	admin If you want to change the username for accessing the webGUI, enter it here.
Password	(confirmation) If you want to change the password for accessing the webGUI, enter it here twice.
webGUI protocol	● HTTP ○ HTTPS
webGUI port	Enter a custom port number for the webGUI above if you want to override the default (80 for HTTP, 443 for HTTPS). Changes will take effect immediately after save.
Time zone	Europe/Berlin Select the location closest to you
Time update interval	300 Minutes between network time sync.; 300 recommended, or 0 to disable
NTP time server	pool.ntp.org Use a space to separate multiple hosts (only one required). Remember to set up at least one DNS server if you enter a host name here!

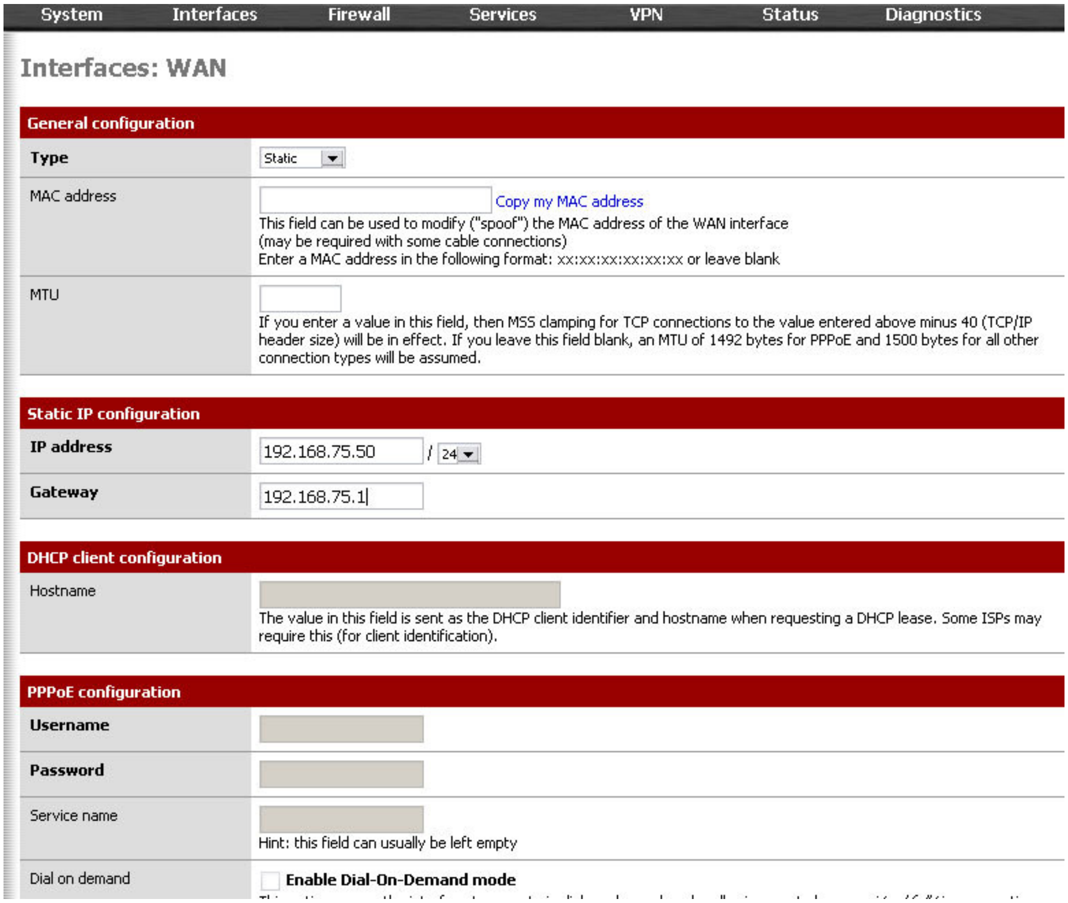
Above you can see my configuration for general setup. As you can see I entered my 3com routers ip-address as dns server, because my router is forwarding dns requests to my ISP. You may have to enter fixed addresses if you have a bigger network with own dns servers.

If you’re layout is more like the second example, pfsense will fetch the dns servers from your ISP over pppoe, just let the “Allow DNS Servers list to be overridden…” box active (this is important!).

As hostname enter something you like the box to be known in you network, if you have an office with an official domain you should give pfsense a FQDN. If you don’t know what a FQDN is, you can most likely just accept “local”.

Adjust the NTP timeserver to the time zone the box will reside.

Next we are going to configure the WAN interface, click on “Interfaces→WAN”:



Interfaces: WAN

General configuration

Type	Static
MAC address	Copy my MAC address This field can be used to modify ("spoof") the MAC address of the WAN interface (may be required with some cable connections) Enter a MAC address in the following format: xx:xx:xx:xx:xx:xx or leave blank
MTU	If you enter a value in this field, then MSS clamping for TCP connections to the value entered above minus 40 (TCP/IP header size) will be in effect. If you leave this field blank, an MTU of 1492 bytes for PPPoE and 1500 bytes for all other connection types will be assumed.

Static IP configuration

IP address	192.168.75.50 / 24
Gateway	192.168.75.1

DHCP client configuration

Hostname	The value in this field is sent as the DHCP client identifier and hostname when requesting a DHCP lease. Some ISPs may require this (for client identification).

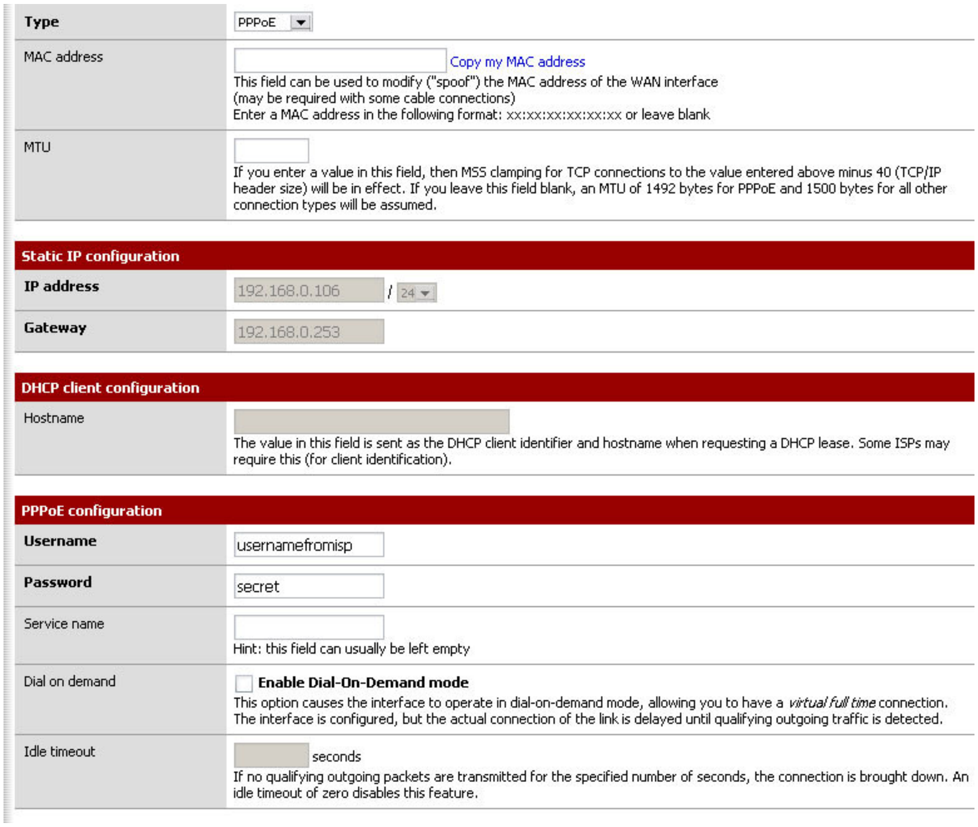
PPPoE configuration

Username	
Password	
Service name	Hint: this field can usually be left empty
Dial on demand	□ **Enable Dial-On-Demand mode**

Again, if you’re layout is like mine you have to use “static” as Type and enter the proper ip-address of your router as gateway. Your WAN ip-address must be on the same subnet as the gateway you entered, in my case that’s 192.168.75.50.

Remove the marker from “Block private networks” if you have a setup like mine, (WAN actually is a private network interface).

For the second layout (pfSense does pppoe) you have to enable pppoe instead of static as Type and enter your account data:



Type	PPPoE ▾
MAC address	Copy my MAC address This field can be used to modify ("spoof") the MAC address of the WAN interface (may be required with some cable connections) Enter a MAC address in the following format: xx:xx:xx:xx:xx:xx or leave blank
MTU	If you enter a value in this field, then MSS clamping for TCP connections to the value entered above minus 40 (TCP/IP header size) will be in effect. If you leave this field blank, an MTU of 1492 bytes for PPPoE and 1500 bytes for all other connection types will be assumed.
Static IP configuration	
IP address	192.168.0.106 / 24 ▾
Gateway	192.168.0.253
DHCP client configuration	
Hostname	The value in this field is sent as the DHCP client identifier and hostname when requesting a DHCP lease. Some ISPs may require this (for client identification).
PPPoE configuration	
Username	usernamefromisp
Password	secret
Service name	Hint: this field can usually be left empty
Dial on demand	□ **Enable Dial-On-Demand mode** This option causes the interface to operate in dial-on-demand mode, allowing you to have a *virtual full time* connection. The interface is configured, but the actual connection of the link is delayed until qualifying outgoing traffic is detected.
Idle timeout	seconds If no qualifying outgoing packets are transmitted for the specified number of seconds, the connection is brought down. An idle timeout of zero disables this feature.

You can block private Networks in this configuration, since your WAN will be official through pppoe.

After this initial configuration we can continue to the next step (press “save”).

Creating the certificates on Unix/Linux (not needed for Site-to-Site)
This is already well described in the existing docs, but I will cover it again. Switch to your favourite Unix/Linux box (FreeBSD is nice btw) and download the OpenVPN source code. We have to create some certificates for our server and for a few clients, I will use the same values and names as the existing doc does, so it may sound familiar if you already read that article (they worked for me). You can also create the certificates on windows, but I never done that so I can’t say how that works.

Download the latest source code from:
http://openvpn.net/download.html

You can directly download it with “fetch” or “wget” if you are on a box without gui.

Untar it with “tar -xvzf openvpn-*.tar.gz” and change to the “easy-rsa” directory. Open the “vars” file with your favourite editor (vi/vim/…) and edit the values at the bottom of the file to your needs. They will be used as defaults in the other scripts, so you don’t have to type them over and over again (this step is not necessary).

```
export KEY_COUNTRY=DE
export KEY_PROVINCE=BA
export KEY_CITY=COLOGNE
export KEY_ORG="Organisation name"
export KEY_EMAIL="me@myhost.mydomain"
```

After that some scripts need to be executed, if asked for “Common Name” enter the hostname you used in “General Setup” this time. Here are my keystrokes:

```
[/tmp/openvpn-2.0.8/easy-rsa]# source ./vars
[/tmp/openvpn-2.0.8/easy-rsa]# ./clean-all
[/tmp/openvpn-2.0.8/easy-rsa]# ./build-ca
.
.
.
Country Name (2 letter code) [DE]: (press enter)
State or Province Name (full name) [BA]:  (press enter)
Locality Name (eg, city) [COLOGNE]): (press enter)
Organization Name (eg, company) [Organisation name]: (press enter)
Organizational Unit Name (eg, section) [ ]:(press enter)
Common Name (eg, your name or your server's hostname)  [ ]:gate.local
Email Address [me@myhost.mydomain]: (press enter)
.
.
.
```

In the next script you have to enter “server” as Common Name.
Say “y” to sign the certificate.

```
[/tmp/openvpn-2.0.8/easy-rsa]# ./build-key-server server
.
.
.
```

Now build the DH parameters:

```
[/tmp/openvpn-2.0.8/easy-rsa]# ./build-dh
.
.
.
```

And finally some certificates for your clients:

```
[/tmp/openvpn-2.0.8/easy-rsa]# ./build-key client1
.
[/tmp/openvpn-2.0.8/easy-rsa]# ./build-key client2
```

```
.
[/tmp/openvpn-2.0.8/easy-rsa]# ./build-key client3
```

You can create as many client certificates as you like, if you later want some more clients, just create new with ./build-key client_name. Always use a different name as parameter. But remember, the other keys you created before must be present in the same directory or it won't work (backup the directory where you created the files, so you can create new client certificates without installing all certificates again).

Creating certificates manually on Windows (by Hernan Maslowski)

You can also create the keys with a Win32 program called "My Certificate Wizard": http://www.openvpn.se/mycert/

Generate the master Certificate Authority (CA) certificate & key

In this section we will generate a master CA certificate/key, a server certificate/key, and certificates/keys for 3 separate clients.

Open up a Command Prompt window and cd to **\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
```

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.

Next, 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:

```
ai:c:\program files\openvpn\easy-rsa\build-ca
Generating a 1024 bit RSA private key
............++++++
...........++++++
writing new private key to '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) [KG]:
State or Province Name (full name) [NA]:
Locality Name (eg, city) [BISHKEK]:
Organization Name (eg, company) [OpenVPN-TEST]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:OpenVPN-CA
Email Address [me@myhost.mydomain]:
```

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, I used "OpenVPN-CA".

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]".

Generate certificates & keys for 3 clients

Generating client certificates is very similar to the previous step.

build-key client1
build-key client2
build-key client3

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

Remember that 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.

Generate Diffie Hellman parameters

Diffie Hellman parameters must be generated for the OpenVPN server.

build-dh

Output:

```
ai:c:\program files\openvpn\easy-rsa\build-dh
```

```
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
..................+.............................................
....................+..............+..................+.........
........................................
```

Key Files

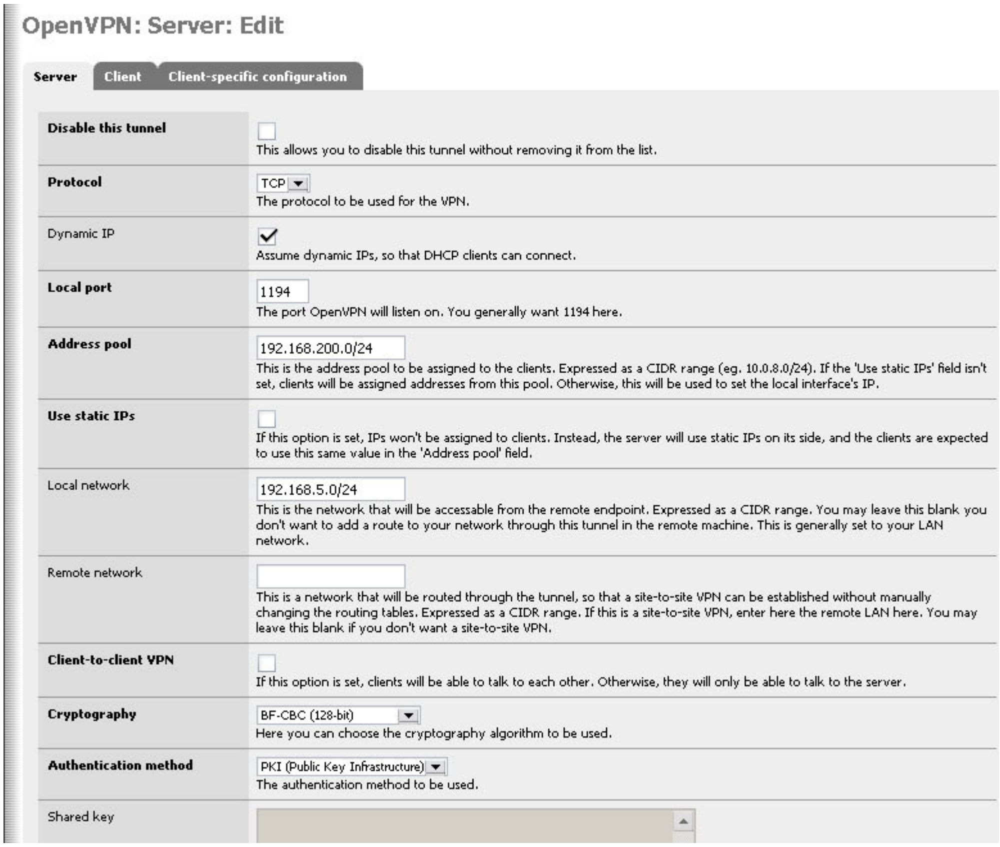
Now we will find our newly-generated keys and certificates in the **keys** subdirectory. Here is an explanation of the relevant files:

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
client3.crt	client3 only	Client3 Certificate	NO
client3.key	client3 only	Client3 Key	YES

We now have all the keys we need, so we can continue to the next step.

Setting up OpenVPN for road warriors (= remote clients)

Ok finally we are ready to configure OpenVPN, click “VPN→OVPN” on the menu and the little “+” box to add a new tunnel.



OpenVPN: Server: Edit

Disable this tunnel	□ This allows you to disable this tunnel without removing it from the list.
Protocol	TCP ▾ The protocol to be used for the VPN.
Dynamic IP	☑ Assume dynamic IPs, so that DHCP clients can connect.
Local port	1194 The port OpenVPN will listen on. You generally want 1194 here.
Address pool	192.168.200.0/24 This is the address pool to be assigned to the clients. Expressed as a CIDR range (eg. 10.0.8.0/24). If the 'Use static IPs' field isn't set, clients will be assigned addresses from this pool. Otherwise, this will be used to set the local interface's IP.
Use static IPs	□ If this option is set, IPs won't be assigned to clients. Instead, the server will use static IPs on its side, and the clients are expected to use this same value in the 'Address pool' field.
Local network	192.168.5.0/24 This is the network that will be accessable from the remote endpoint. Expressed as a CIDR range. You may leave this blank you don't want to add a route to your network through this tunnel in the remote machine. This is generally set to your LAN network.
Remote network	 This is a network that will be routed through the tunnel, so that a site-to-site VPN can be established without manually changing the routing tables. Expressed as a CIDR range. If this is a site-to-site VPN, enter here the remote LAN here. You may leave this blank if you don't want a site-to-site VPN.
Client-to-client VPN	□ If this option is set, clients will be able to talk to each other. Otherwise, they will only be able to talk to the server.
Cryptography	BF-CBC (128-bit) ▾ Here you can choose the cryptography algorithm to be used.
Authentication method	PKI (Public Key Infrastructure) ▾ The authentication method to be used.
Shared key	

As you can see I used “TCP” for protocol since UDP is known to be filtered badly by some routers. Using TCP is a bit slower, but safer for now.

Click on “Dynamic IP”, we want to allow remote clients with dial-in addresses to be able to connect to our server (the typical “road warrior”).

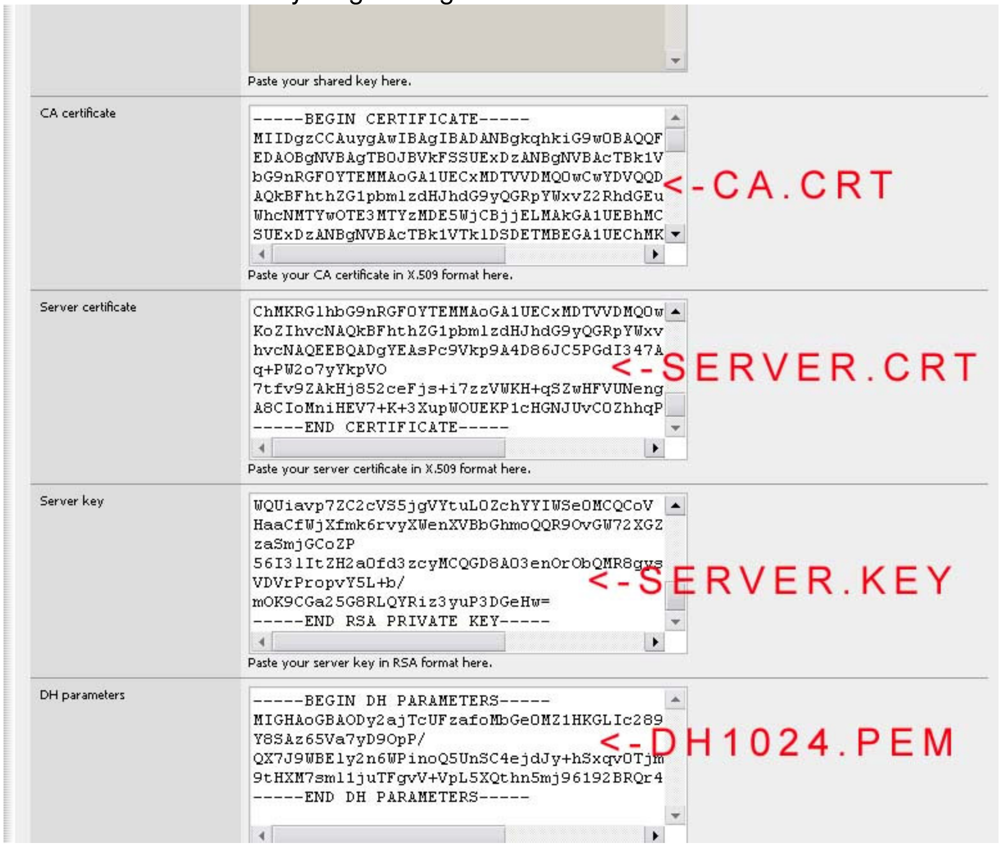
“Address pool” must be an independent subnet you are not using anywhere else. This is important! Do not put the same subnet in here as you entered for your LAN or WAN connection. In my case I used “192.168.200.0/24”.

Change “Authentication method” to PKI.

Now we have to cut & paste our keys, use your favourite editor to open the files in plain text (you can use “cat filename” on Unix/Linux or Notepad on Windows). Always include the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----

parts and everything between them.

Take a look where everything belongs:



Paste your shared key here.

CA certificate

```
-----BEGIN CERTIFICATE-----
MIIDgzCCAuygAwIBAgIBADANBgkqhkiG9w0BAQQF
EDAOBgNVBAgTBOJBVkFSSUExDzANBgNVBAcTBk1V
bG9nRGF0YTEMMAoGA1UECxMDTVVDMQ0wCwYDVQQD
AQkBFhthZG1pbmlzdHJhdG9yQGRpYWxvZ2RhdGEu
WhcNMTYwOTE3MTYzMDE5WjCBjjELMAkGA1UEBhMC
SUExDzANBgNVBAcTBk1VTk1DSDETMBEGA1UEChMK
```

<-CA.CRT

Paste your CA certificate in X.509 format here.

Server certificate

```
ChMKRG1hbG9nRGF0YTEMMAoGA1UECxMDTVVDMQ0w
KoZIhvcNAQkBFhthZG1pbmlzdHJhdG9yQGRpYWxv
hvcNAQEEBQADgYEAsPc9Vkp9A4D86JC5PGdI347A
q+PW2o7yYkpVO
7tfv9ZAkHj852ceFjs+i7zzVWKH+qSZwHFVUNeng
A8CIoMniHEV7+K+3XupWOUEKP1cHGNJUvCOZhhqP
-----END CERTIFICATE-----
```

<-SERVER.CRT

Paste your server certificate in X.509 format here.

Server key

```
WQUiavp7ZC2cVS5jgVYtuL0ZchYYIWSe0MCQCoV
HaaCfWjXfmk6rvyXWenXVBbGhmoQQR9OvGW72XGZ
zaSmjGCoZP
56I31ItZH2a0fd3zcyMCQGD8A03enOrObQMR8gvs
VDVrPropvY5L+b/
mOK9CGa25G8RLQYRiz3yuP3DGeHw=
-----END RSA PRIVATE KEY-----
```

<-SERVER.KEY

Paste your server key in RSA format here.

DH parameters

```
-----BEGIN DH PARAMETERS-----
MIGHAoGBAODy2ajTcUFzafoMbGe0MZ1HKGLIc289
Y8SAz65Va7yD9OpP/
QX7J9WBEly2n6WPinoQ5UnSC4ejdJy+hSxqv0Tjm
9tHXM7sml1juTFgvV+VpL5XQthn5mj96192BRQr4
-----END DH PARAMETERS-----
```

<-DH1024.PEM

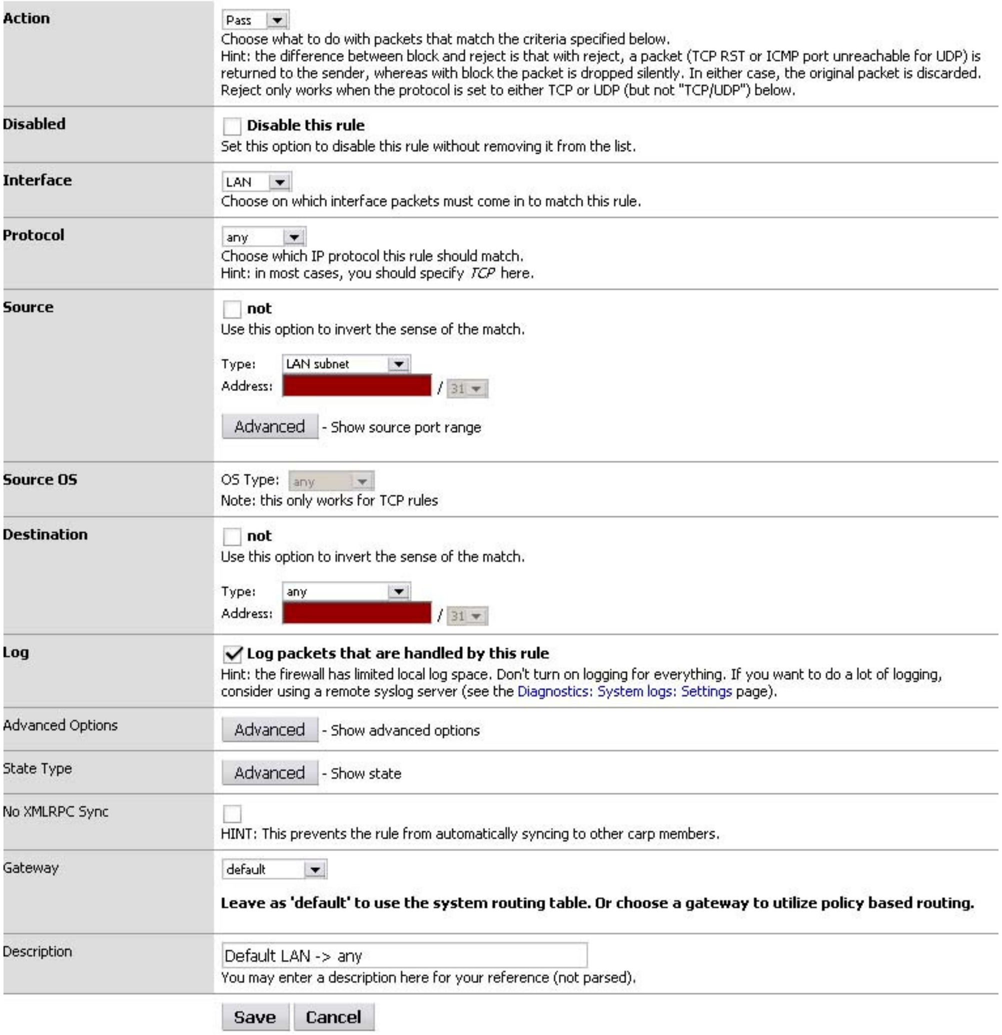
After the certificates are in place, deactivate LZO-Compression (for testing, if everything worked you can activate it and change the value in the client file as described later). Write something useful in the “Description” field, like “Road Warrior OVPN”. As usual click “Save”.

If you want to allow your road warriors to connect to other subnets as the LAN interface (for example DMZ), you have to push a route in the “Custom options” field. If your DMZ is 192.168.100.0/24 for example, you have to write the following to route your road warriors to DMZ:
push “route 192.168.100.0 255.255.255.0”

Firewall Rules

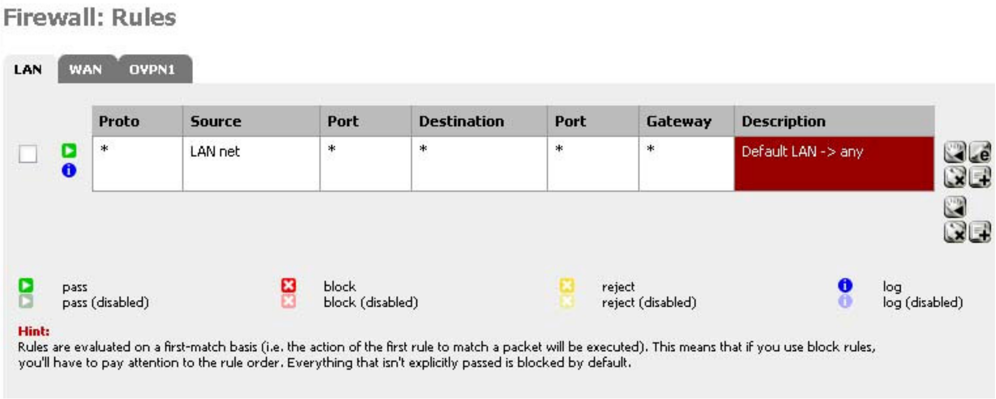
We need some very basic firewall rules to get it running first, later if you are sure that everything works as desired insert more complex rules if you need to.

On the menu click on “Firewall→Rules”, then click on the LAN button.
In my case the rule to allow LAN traffic to anything already existed, if it does not in your configuration click on the little “+” box an edit the values as shown:



Action	Pass ▾ Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded. Reject only works when the protocol is set to either TCP or UDP (but not "TCP/UDP") below.
Disabled	□ **Disable this rule** Set this option to disable this rule without removing it from the list.
Interface	LAN ▾ Choose on which interface packets must come in to match this rule.
Protocol	any ▾ Choose which IP protocol this rule should match. Hint: in most cases, you should specify *TCP* here.
Source	□ **not** Use this option to invert the sense of the match. Type: LAN subnet ▾ Address: / 31 ▾ Advanced - Show source port range
Source OS	OS Type: any ▾ Note: this only works for TCP rules
Destination	□ **not** Use this option to invert the sense of the match. Type: any ▾ Address: / 31 ▾
Log	☑ **Log packets that are handled by this rule** Hint: the firewall has limited local log space. Don't turn on logging for everything. If you want to do a lot of logging, consider using a remote syslog server (see the Diagnostics: System logs: Settings page).
Advanced Options	Advanced - Show advanced options
State Type	Advanced - Show state
No XMLRPC Sync	□ HINT: This prevents the rule from automatically syncing to other carp members.
Gateway	default ▾ **Leave as 'default' to use the system routing table. Or choose a gateway to utilize policy based routing.**
Description	Default LAN -> any You may enter a description here for your reference (not parsed).
	Save Cancel

Logging is not necessary but good for testing. The rule should now look like this:



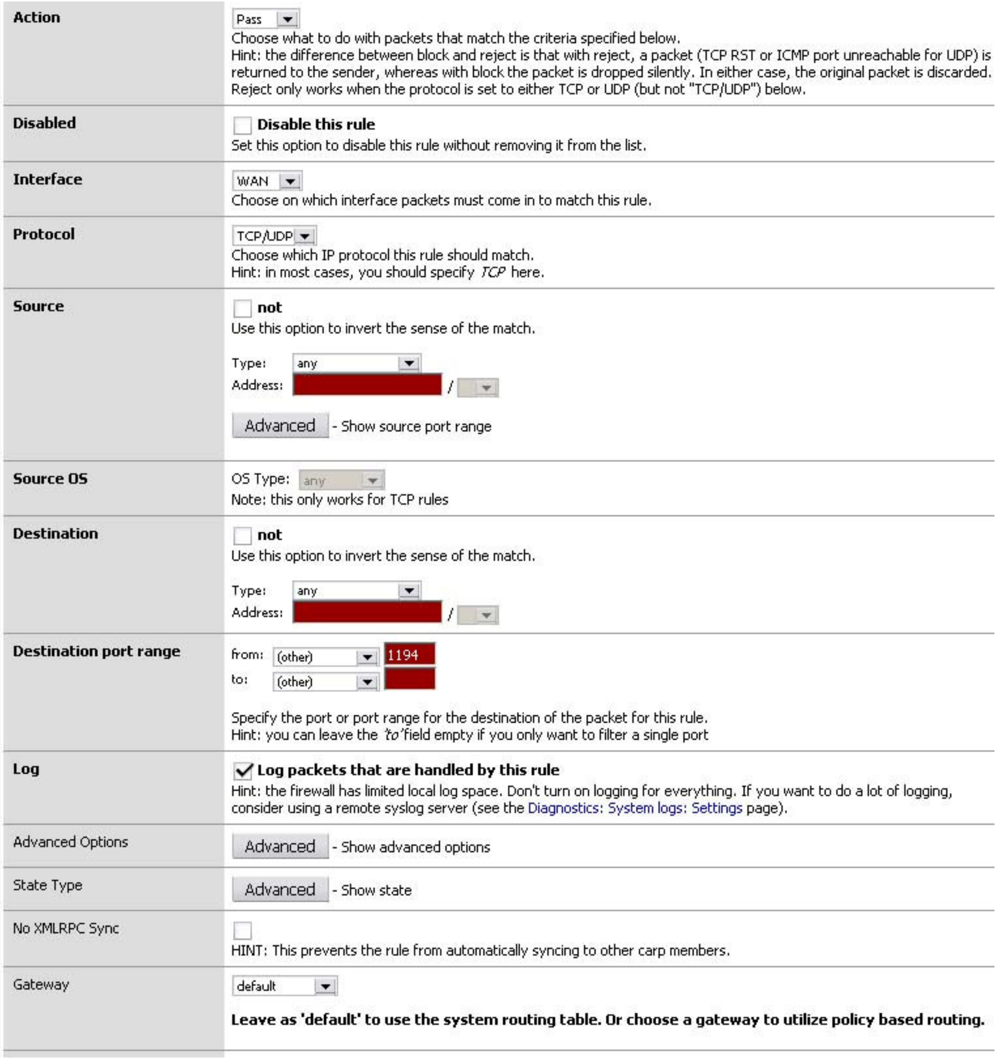
Firewall: Rules

		Proto	Source	Port	Destination	Port	Gateway	Description
□	▶ ℹ	*	LAN net	*	*	*	*	Default LAN -> any

🟩 pass	🟥 block	🟨 reject	ℹ log
pass (disabled)	block (disabled)	reject (disabled)	log (disabled)

Hint:
Rules are evaluated on a first-match basis (i.e. the action of the first rule to match a packet will be executed). This means that if you use block rules, you'll have to pay attention to the rule order. Everything that isn't explicitly passed is blocked by default.

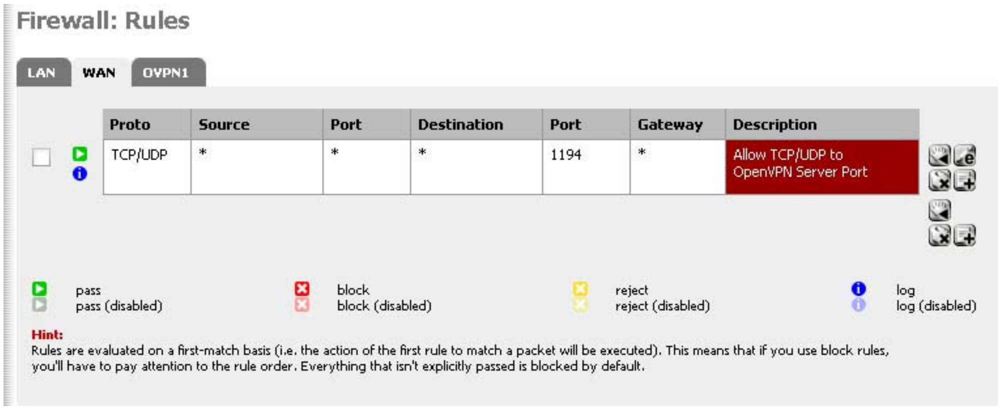
Next click on the WAN tab and on the little "+" box to add a new rule:



Action	Pass ▾ Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded. Reject only works when the protocol is set to either TCP or UDP (but not "TCP/UDP") below.
Disabled	□ **Disable this rule** Set this option to disable this rule without removing it from the list.
Interface	WAN ▾ Choose on which interface packets must come in to match this rule.
Protocol	TCP/UDP ▾ Choose which IP protocol this rule should match. Hint: in most cases, you should specify *TCP* here.
Source	□ **not** Use this option to invert the sense of the match. Type: any ▾ Address: / ▾ Advanced - Show source port range
Source OS	OS Type: any ▾ Note: this only works for TCP rules
Destination	□ **not** Use this option to invert the sense of the match. Type: any ▾ Address: / ▾
Destination port range	from: (other) ▾ 1194 to: (other) ▾ Specify the port or port range for the destination of the packet for this rule. Hint: you can leave the *'to'* field empty if you only want to filter a single port
Log	☑ **Log packets that are handled by this rule** Hint: the firewall has limited local log space. Don't turn on logging for everything. If you want to do a lot of logging, consider using a remote syslog server (see the Diagnostics: System logs: Settings page).
Advanced Options	Advanced - Show advanced options
State Type	Advanced - Show state
No XMLRPC Sync	□ HINT: This prevents the rule from automatically syncing to other carp members.
Gateway	default ▾ **Leave as 'default' to use the system routing table. Or choose a gateway to utilize policy based routing.**

As Protocol enter "TCP/UDP" while we are testing, later edit the rule to only match the protocol you finally used.

"Destination port range" is the port our OpenVPN Server is listening, by default this is 1194. The rule should look like this after you saved:



Firewall: Rules

		Proto	Source	Port	Destination	Port	Gateway	Description
□	▶ ●	TCP/UDP	*	*	*	1194	*	Allow TCP/UDP to OpenVPN Server Port

🟩 pass
🟩 pass (disabled)

🟥 block
🟥 block (disabled)

🟨 reject
🟨 reject (disabled)

● log
● log (disabled)

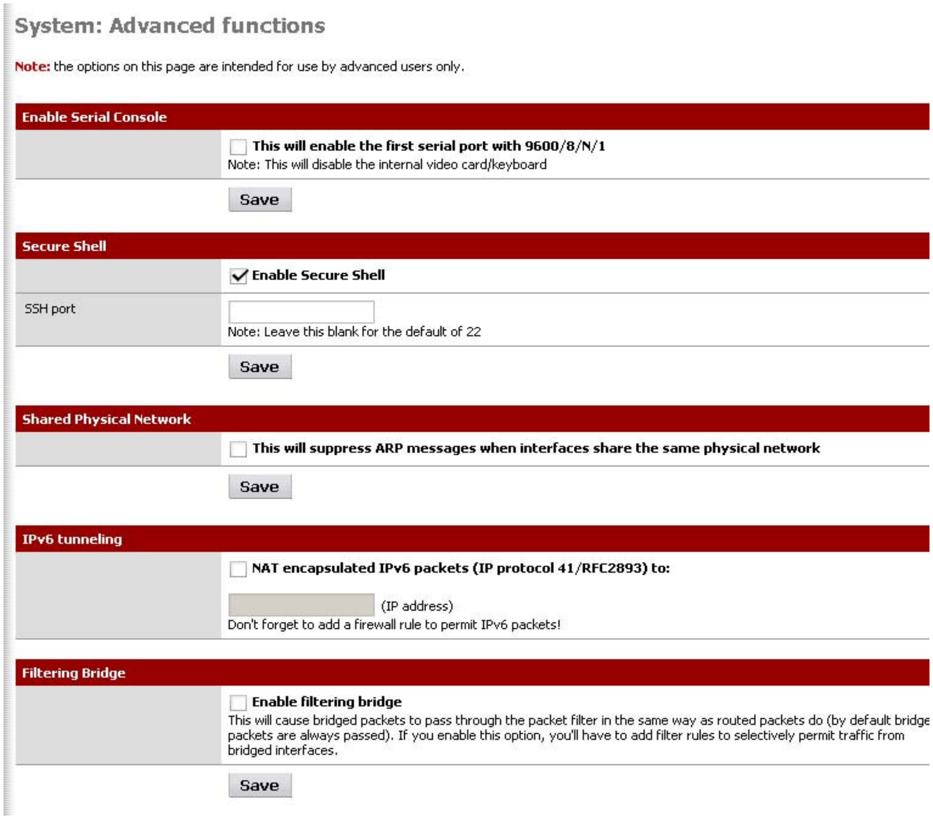
Hint:
Rules are evaluated on a first-match basis (i.e. the action of the first rule to match a packet will be executed). This means that if you use block rules, you'll have to pay attention to the rule order. Everything that isn't explicitly passed is blocked by default.

Ok, primary objectives completed ;)

Interfaces, OpenVPN and firewall rules configured, we just need to save the configuration to floppy and reboot to see if everything comes up smoothly.

Save configuration to floppy disk

If you have a monitor and keyboard on your pfsense box, you can directly choose the correct option as shown later, if not like in my case you have to enable the ssh-server. Click on "Setup→Advanced" in the menu and enable "Secure Shell" as shown below:



System: Advanced functions

Note: the options on this page are intended for use by advanced users only.

Enable Serial Console

☐ **This will enable the first serial port with 9600/8/N/1**
Note: This will disable the internal video card/keyboard

Save

Secure Shell

☑ **Enable Secure Shell**

SSH port: ___
Note: Leave this blank for the default of 22

Save

Shared Physical Network

☐ **This will suppress ARP messages when interfaces share the same physical network**

Save

IPv6 tunneling

☐ **NAT encapsulated IPv6 packets (IP protocol 41/RFC2893) to:**

___ (IP address)
Don't forget to add a firewall rule to permit IPv6 packets!

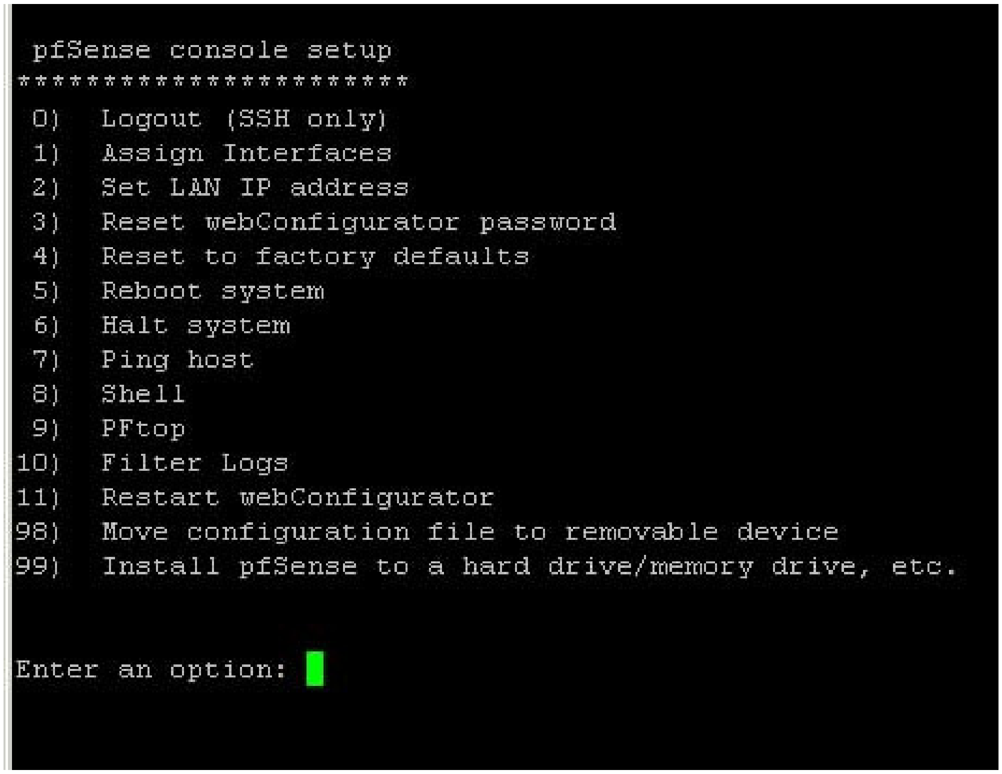
Filtering Bridge

☐ **Enable filtering bridge**
This will cause bridged packets to pass through the packet filter in the same way as routed packets do (by default bridge packets are always passed). If you enable this option, you'll have to add filter rules to selectively permit traffic from bridged interfaces.

Save

Finally you can use your favourite ssh-client to connect to pfsense (use the LAN interface ip-address), on Windows you can use putty. If asked for username and password enter "admin" and "pfsense" (change that later!!!).

You will see a screen like this:



```
 pfSense console setup
***********************
 0)   Logout (SSH only)
 1)   Assign Interfaces
 2)   Set LAN IP address
 3)   Reset webConfigurator password
 4)   Reset to factory defaults
 5)   Reboot system
 6)   Halt system
 7)   Ping host
 8)   Shell
 9)   PFtop
10)   Filter Logs
11)   Restart webConfigurator
98)   Move configuration file to removable device
99)   Install pfSense to a hard drive/memory drive, etc.


Enter an option: █
```

Enter “98” and press enter, answer the next question with “fd0” and press enter again, the configuration will be saved to floppy disk. If “98” is not shown in your menu, you maybe forgot to insert the floppy disk *before* booting from cdrom. After the backup is finished press “5” and enter to reboot pfsense.

Let pfsense reboot, if everything worked well, you should be able to connect to the web interface as you did before (pfsense will automatically read the configuration from the floppy disk while booting).

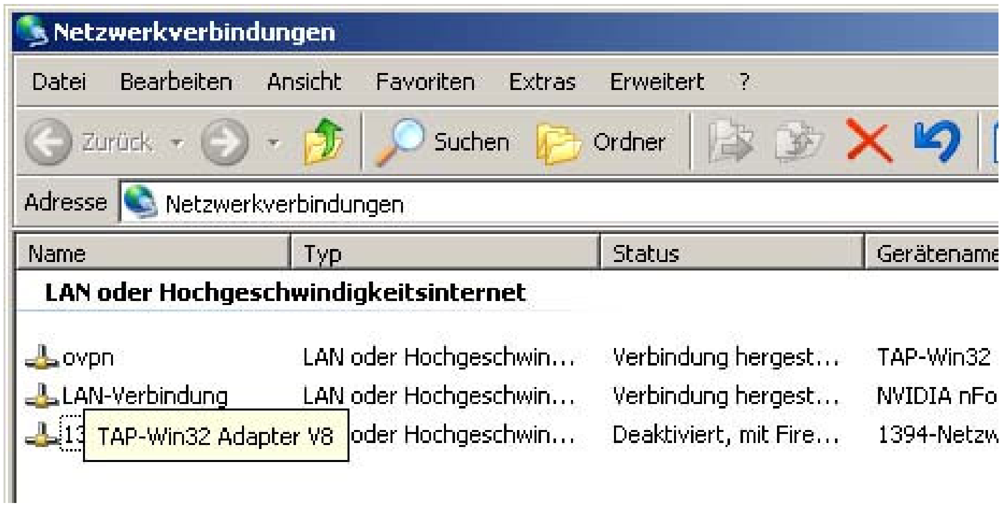
Time to move on to the client configuration, since most people using linux/unix know what they do I will only cover the Windows version (values are the same anyway, just cut & paste).

Road Warrior configuration on Windows

Now we are going to configure a typical client that will be able to connect from different dial-in networks from around the world. On windows make sure you have sufficient permissions to change routes and install new interfaces, you most likely need “administrator” privileges to do so (I am not sure here).

Download the OpenVPN client from http://openvpn.se and install it.
The client installs a new network interface with a long winded name, which we need to rename to “ovpn” or something like that (just short and no spaces).

Take a look:

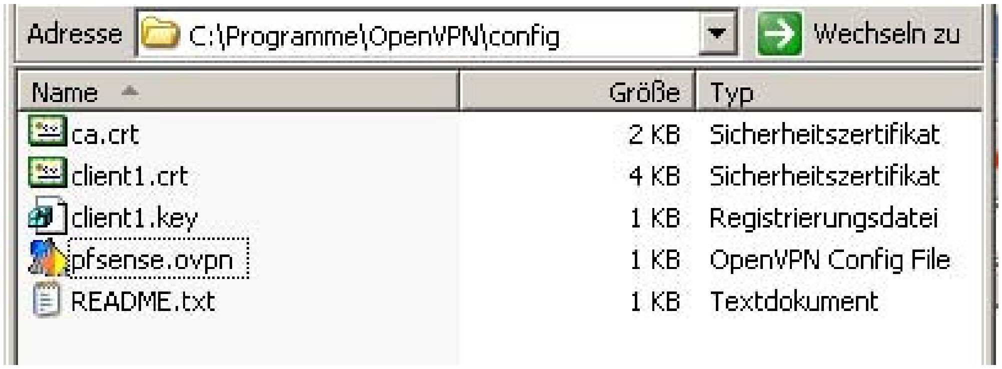


Name	Typ	Status	Gerätename
LAN oder Hochgeschwindigkeitsinternet			
ovpn	LAN oder Hochgeschwin...	Verbindung hergest...	TAP-Win32
LAN-Verbindung	LAN oder Hochgeschwin...	Verbindung hergest...	NVIDIA nFo
13	oder Hochgeschwin...	Deaktiviert, mit Fire...	1394-Netzw

Now, create a new text file in “c:\program files\openvpn\config folder” (or wherever you installed it) named “pfsense.ovpn” (you may change pfsense to whatever you like to describe the tunnel, but keep the ending). Copy & paste the following configuration:

```
float
port 1194
dev tun
dev-node ovpn
proto tcp-client
remote yourpfsensebox 1194
ping 10
persist-tun
persist-key
tls-client
ca ca.crt
cert client1.crt
key client1.key
ns-cert-type server
#comp-lzo ← to enable LZO remove the #
pull
verb 4
```

dev-node **ovpn** must match the name of the renamed interface, **yourpfsensebox** the ip-adress of your pfsense box (or router that does NAT like in my case). Remember the client certificates? We need to copy some them over to the same directory, for the first client copy “ca.crt”, “client1.crt” and “client1.key”. You always need “ca.crt” and the proper client files.



Adresse C:\Programme\OpenVPN\config Wechseln zu

Name	Größe	Typ
ca.crt	2 KB	Sicherheitszertifikat
client1.crt	4 KB	Sicherheitszertifikat
client1.key	1 KB	Registrierungsdatei
pfsense.ovpn	1 KB	OpenVPN Config File
README.txt	1 KB	Textdokument

Now take a deep breath and right click the “pfsense.ovpn” file and choose “Start OpenVPN on this configuration file”, the client will try connecting to your pfsense box (you should test this from an external network; I did it from home→office).

If the tunnel connect succeeded (“Initialization sequence completed” should be the last log on your shell with some “RRWWRrwrw” stuff following) and you cannot ping the internal hosts, keep in mind that you have to set pfsense as default gateway on all LAN servers you want to be able to connect to.

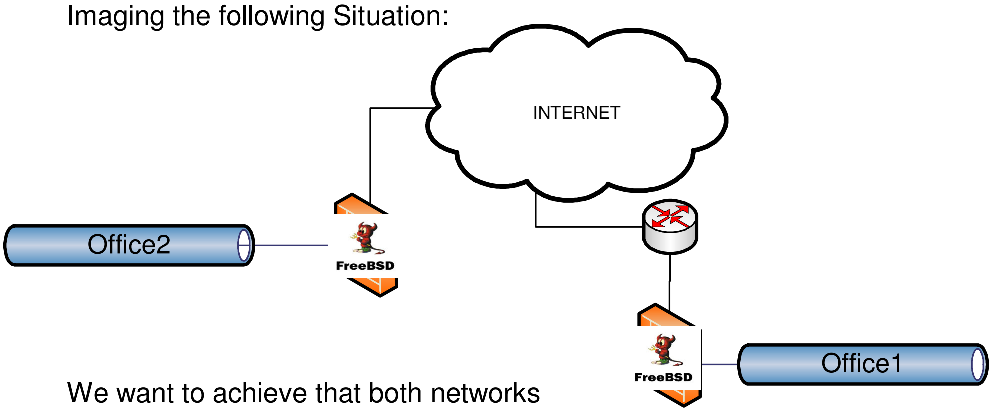
That’s it; the tunnel should now be ready to serve…

Setting up Site-to-Site OpenVPN

In this section I am going to explain how to setup pfsense with OpenVPN to connect two sites with a persistent tunnel.

Office1 LAN: 192.168.0.0/24
Office2 LAN: 192.168.1.0/24

Imaging the following Situation:

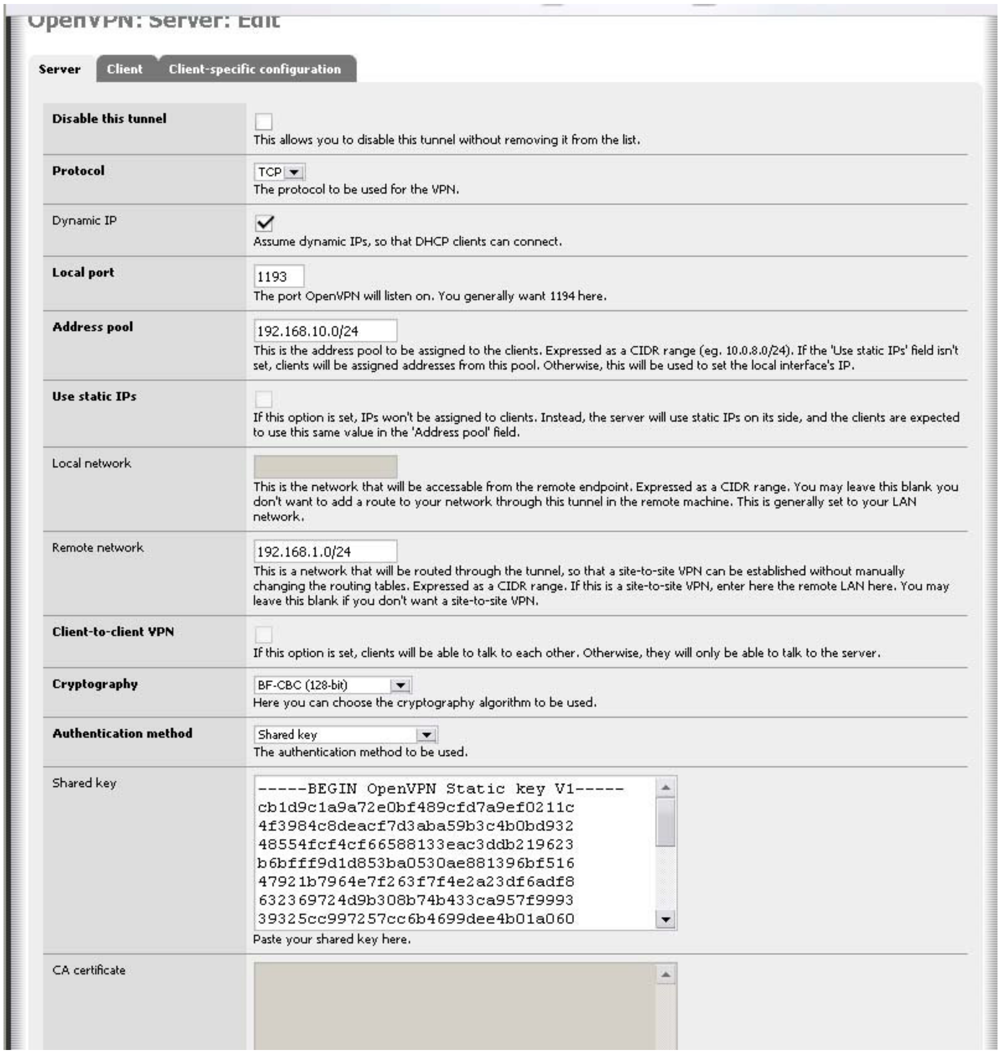


We want to achieve that both networks “feel” like the same, so that every host from Office1 can connect to every host on Office2 and vice versa.

Setup pfsense as described above (leave out the road warrior stuff if you don’t want support for remote clients). If both sites can access the internet, the following configuration should connect them.

If you have already configured pfsense for road warriors you have to setup an additional tunnel, don’t change the existing configuration!

<u>Office1:</u>
We will now configure “Office1” as the Server.
Click “VPN→ OpenVPN” and add a new tunnel by clicking the little “+” box:



OpenVPN: Server: Edit

Disable this tunnel	□ This allows you to disable this tunnel without removing it from the list.
Protocol	TCP The protocol to be used for the VPN.
Dynamic IP	☑ Assume dynamic IPs, so that DHCP clients can connect.
Local port	1193 The port OpenVPN will listen on. You generally want 1194 here.
Address pool	192.168.10.0/24 This is the address pool to be assigned to the clients. Expressed as a CIDR range (eg. 10.0.8.0/24). If the 'Use static IPs' field isn't set, clients will be assigned addresses from this pool. Otherwise, this will be used to set the local interface's IP.
Use static IPs	□ If this option is set, IPs won't be assigned to clients. Instead, the server will use static IPs on its side, and the clients are expected to use this same value in the 'Address pool' field.
Local network	 This is the network that will be accessable from the remote endpoint. Expressed as a CIDR range. You may leave this blank you don't want to add a route to your network through this tunnel in the remote machine. This is generally set to your LAN network.
Remote network	192.168.1.0/24 This is a network that will be routed through the tunnel, so that a site-to-site VPN can be established without manually changing the routing tables. Expressed as a CIDR range. If this is a site-to-site VPN, enter here the remote LAN here. You may leave this blank if you don't want a site-to-site VPN.
Client-to-client VPN	□ If this option is set, clients will be able to talk to each other. Otherwise, they will only be able to talk to the server.
Cryptography	BF-CBC (128-bit) Here you can choose the cryptography algorithm to be used.
Authentication method	Shared key The authentication method to be used.
Shared key	-----BEGIN OpenVPN Static key V1----- cb1d9c1a9a72e0bf489cfd7a9ef0211c 4f3984c8deacf7d3aba59b3c4b0bd932 48554fcf4cf66588133eac3ddb219623 b6bfff9d1d853ba0530ae881396bf516 47921b7964e7f263f7f4e2a23df6adf8 632369724d9b308b74b433ca957f9993 39325cc997257cc6b4699dee4b01a060 Paste your shared key here.
CA certificate	

Change protocol to TCP, if you have additional tunnels remember to use another port for this one, I used 1193 since 1194 is already used by my road warrior tunnel.

“Address pool” must be a network you do not use anywhere else, I used 192.168.10.0/24 since neither Office1 nor Office2 use this subnet.

For “Remote network” enter the LAN subnet of Office2 (remember, we do configure this on Office1) so we want 192.168.1.0/24 in here.

Now we will create the “Shared key”.
Login to your pfsense on Office1 over SSH and type “8” for the shell and type in the following command:

```
# openvpn --genkey --secret shared.key
```

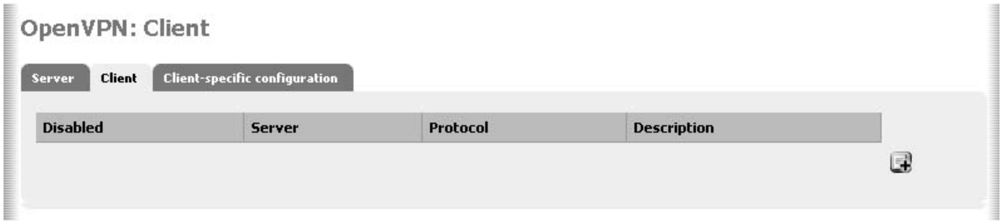
This command will create the shared key for this OpenVPN server.
Now ‘cat’ the file and cut&paste it to the webgui like shown above.

```
# cat shared.key
```

Press “Save”. Reboot the box to see if everything comes up well.
Now copy the shared key you used to an USB-Stick or find some other way to transfer it safely to Office2 (Email is a bad idea).

So, first part is done, we have configured the “listening” tunnel, remember to create a corresponding firewall rule to allow TCP traffic to port 1193 on your WAN interface (check the road warrior section how that is done).

Office2:
On Office2 we will configure the client side of the tunnel, click “VPN→OpenVPN→Client” as shown here:



OpenVPN: Client

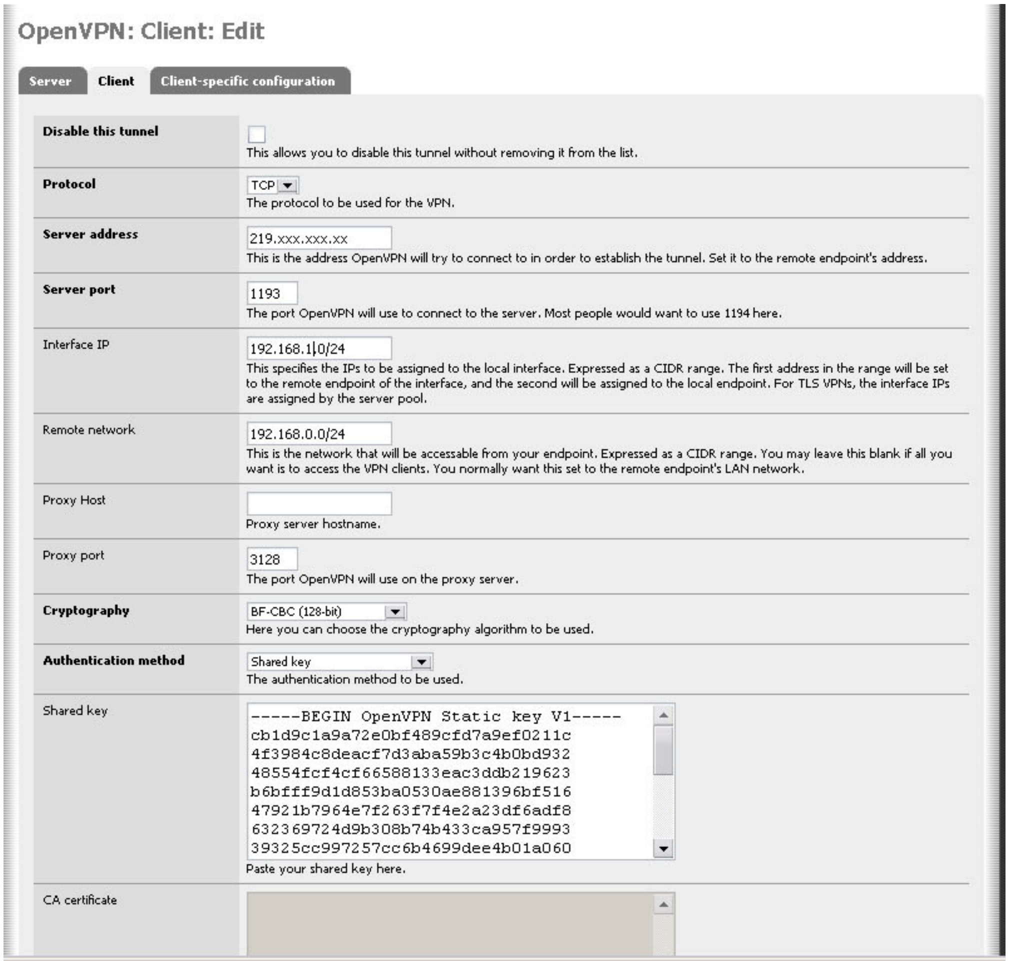
Disabled	Server	Protocol	Description

And again the little “+” box to add a tunnel.

Prepare the shared key you created on Office1, we will need it now.

Set Protocol to “TCP”, “Server address” must be set to the official IP of Office1 (if that’s not the WAN interface of pfsense, your router has to do portforwarding), “Server port” is 1193. “Interface IP” should be filled with your local subnet.
The “Remote network” field is the LAN subnet of Office1.

Now paste the shared key from Office1 to the appropriate field. Click “Save” and reboot the box to see if everything works fine.



OpenVPN: Client: Edit

Disable this tunnel	□ This allows you to disable this tunnel without removing it from the list.
Protocol	TCP The protocol to be used for the VPN.
Server address	219.xxx.xxx.xx This is the address OpenVPN will try to connect to in order to establish the tunnel. Set it to the remote endpoint's address.
Server port	1193 The port OpenVPN will use to connect to the server. Most people would want to use 1194 here.
Interface IP	192.168.1.0/24 This specifies the IPs to be assigned to the local interface. Expressed as a CIDR range. The first address in the range will be set to the remote endpoint of the interface, and the second will be assigned to the local endpoint. For TLS VPNs, the interface IPs are assigned by the server pool.
Remote network	192.168.0.0/24 This is the network that will be accessable from your endpoint. Expressed as a CIDR range. You may leave this blank if all you want is to access the VPN clients. You normally want this set to the remote endpoint's LAN network.
Proxy Host	 Proxy server hostname.
Proxy port	3128 The port OpenVPN will use on the proxy server.
Cryptography	BF-CBC (128-bit) Here you can choose the cryptography algorithm to be used.
Authentication method	Shared key The authentication method to be used.
Shared key	-----BEGIN OpenVPN Static key V1----- cb1d9c1a9a72e0bf489cfd7a9ef0211c 4f3984c8deacf7d3aba59b3c4b0bd932 48554fcf4cf66588133eac3ddb219623 b6bfff9d1d853ba0530ae881396bf516 47921b7964e7f263f7f4e2a23df6adf8 63236972 4d9b308b74b433ca957f9993 39325cc997257cc6b4699dee4b01a060 Paste your shared key here.
CA certificate	

The tunnel between both sites should no be up and running. Try to ping hosts from each others subnet to verify.

What now?

All what is left is to harden the firewall rules and to enable LZO-compression if you like to. You can switch to UDP if speed matters, just change the values for tunnel, WAN firewall rule and client configuration file. Remember, whatever you change you have to do that on the server and client side. If you need to grant access to your DMZ or other interfaces you have to "push" routes in your tunnel configuration.

FAQ

In this section I will add solutions to common problems that may still exist after following this tutorial.

1. I can't run the commands to create OpenVPN Keys

Most likely you are using an incompatible shell, try installing "bash".
You can also create the keys with a Win32 program called "My Certificate Wizard": http://www.openvpn.se/mycert/

2. How can I access "Windows Shares" without bridging?

To access "Windows Shares" you have to install a WINS server in your network and configure all servers with shares to use it (use TCP as protocol). In your tunnel configuration in the field "Custom options" you have to enter the internal DNS server(s) and your WINS box. A valid string could look like this one (you can enter more push options; just separate them with a semicolon):

push "dhcp-option DNS 192.168.0.130";push "dhcp-option WINS 192.168.0.131"

3. Windows Firewall warning!

The Windows Firewall has problems with OpenVPN, so deactivate the firewall on your tun/tap interface if you have problems connecting.
Generally you should deactivate all Desktop-Firewalls while testing.

Have fun with this great product!

Gino Thomas
http://www.uplinksecurity.de
thomas0@fhm.edu