OPENVPN - HARDENED MULTI-CLIENT SETUP

Openvpn is truly a brilliant open-source product, it is a VPN solution for almost every solution and need. Now I know we have had a previous article on this product (done by Nic Maurel, see here), but recently I found myself having to do something a little different with it. I had to setup a single VPN server capable of dealing with multiple clients, using TCP and using a very strict and secure setup. So in this article I will be covering off on that type of setup.

What You Need
First you will need the source for *openvpn*. Once you have downloaded and extracted it, you will do the normal *configure, make, make install* dance. You may find yourself need the *lzo* libraries, you can get them *here*. Now once all this has happened, you should have the *openvpn* executable sitting in */usr/local/sbin/openvpn*. Now I generally create the folder */etc/openvpn*, and I am going to assume -for the purposes of this article- that you have the unextracted *openvpn* source in */admin/openvpn*. Also remember that both the clients and the server will need *openvpn* installed.

Certificates and Keys
Now *openvpn* will require you to use certificates if you setup a multi-client setup, lucky for us, they have made it very simple. First we setup our own CA (Certificate Authority)...

```
cd /admin/openvpn/examples/easy-rsa
. ./vars
./clean-all
./build-ca
```

The last command will start an interactive session for setting up the CA certificate, the only thing you really need to worry about here is the *Common Name* entry. Now we create a server key..

```
./build-key-server server
```

Again, the only entry to worry about here is the *Common Name* entry, and in this field put *server*. At the end you will be asked 2 *y/n* questions, for both of them answer *y* (yes). You have now setup your *ca*, and the *server* certificate. Now we will create a certificate for the *USA* office. Adding certificates is easy, just change the name with the command, also remember that all the options should remain default except for the *Common Name* (in our case it is *usa*) which must be the same as the name given to the command..

```
./build-key usa
```

Next we have to build the Diffie-Hellman parameters used by the VPN..

```
./build-dh
```

This will create a file called *dh1024.pem*. Now in the *keys* folder in the *easy-rsa* folder is where all your keys are stored. We will move the server ones to */etc/openvpn*..

```
cd /admin/openvpn/examples/easy-rsa/keys
cp ./ca.crt /etc/openvpn
cp ./dh1024.pem /etc/openvpn
cp ./server.crt /etc/openvpn
cp ./server.key /etc/openvpn
```

Now we will also generate a secret key which we will use in the authentication process..

```
openvpn --genkey --secret /etc/openvpn/key.txt
```

Now we need to copy the files to the client which the client needs..

```
scp /admin/openvpn/examples/easy-rsa/keys/ca.crt root@usa:/etc/openvpn
scp /admin/openvpn/examples/easy-rsa/keys/usa.* root@usa:/etc/openvpn
```

```
scp /etc/openvpn/key.txt root@usa:/etc/openvpn
```

Now the very last thing we do before creating the configuration files and testing is to ensure that the clients can talk to the server via *TCP* port *1194*. Check your firewall settings to ensure this.

Configuration- Server
The configuration files are not hugely long, but they are very important to get right. Now on the server machine we will create a file called */etc/openvpn/config.vpn*. We will make sure it looks like..

```
local x.x.x.x   ##This sets the ip address the server will listen on
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist /etc/openvpn/ipp.txt
keepalive 60 180
dev tap     ##We will be using TAP devices
proto tcp-server   ##Used to specify the usage of TCP
ca /etc/openvpn/ca.crt
cert /etc/openvpn/server.crt
key /etc/openvpn/server.key
dh /etc/openvpn/dh1024.pem
tls-server
tls-auth /etc/openvpn/key.txt 0
cipher BF-CBC
keysize 448  ##Here we tell Blowfish to use 448 bit encryption
log /var/log/openvpn.log   ##Here we set the logfile
status /var/log/openvpn.status 120  ##This is a status file of the current running
writepid /etc/openvpn/openvpn.pid
link-mtu 1575
comp-lzo
verb 3
mute 10
daemon
```

Configuration- Client
The client configuration file (I've named it */etc/openvpn/config.vpn* as well) looks a similar, but there are some important differences..

```
client      ##This tells it that it is a client
remote x.x.x.x     ##Here you specify the server's IP
dev tap
proto tcp-client
ca /etc/openvpn/ca.crt
cert /etc/openvpn/London.crt
key /etc/openvpn/London.key
tls-client   ##Specifies that it is the TLS client
tls-auth /etc/openvpn/key.txt 1
ns-cert-type server   ##Does server verification
cipher BF-CBC
keysize 448
log /var/log/openvpn.log
status /var/log/openvpn.status 120
writepid /etc/openvpn/openvpn.pid
link-mtu 1575
comp-lzo
verb 3
mute 10
daemon
```

Starting the VPN
Starting up each side is as simple as running the following on each machine..

```
openvpn --config /etc/openvpn/config.vpn
```

Check your log files for errors, but most problems will revolve around the client been able to connect to the listening port on the server. To start the VPN up automatically just put the commands into your startup scripts. Remember that if you want to add a client, just create the certificates, copy the files across, and setup the configuration file.

Final Words

Openvpn truly is a piece of software that is very useful, it can not only encapsulate an entire networks traffic, but it can also travel across a firewall doing address translations. Never mind the fact that you lock it down quite severely. The above configuration works for me and is fairly secure, but there are many other options to the *openvpn* software, like inbuilt routing options, so take a look under the hood. As always have fun and learn.