Using Linux client for Math VPN

Šarūnas Burdulis
Ver. 1, July 2005
Ver. 1.1, 2005.09.01: changed MTU size

Contents

1 Introduction

This mini-HOWTO describes procedures for setting-up and using Linux client to connect to the OpenVPN service running at Math Department. OpenVPN is different from an IPsec-type VPN and has better chances of getting through firewalls and NAT routers. See [1] and [2] for more on how OpenVPN is different from other VPN technologies.

As of March 2005 this HOWTO applies to Linux kernels 2.4/2.6 and OpenVPN 2.x. It is somewhat Debian-oriented, but apart from `apt-get` should be usable with any Linux distribution.

2 Quick start

1. Install OpenVPN:
   `# apt-get install openvpn`
2. Have the following in place:
   `/etc/openvpn/mydccert.pem` — your Dartmouth certificate,
   `/etc/openvpn/mydckey.pem` — your corresponding private key,
   `/etc/ssl/certs/collegeca.pem` — Dartmouth Certificate Authority (CA) certificate.
3. Get Math VPN client configuration file:
   `# scp <`*you*`>@gauss:/usr/local/share/openvpn/mathvpn.conf /etc/openvpn/`

4. Start/stop:
```
# /etc/init.d/openvpn start
# /etc/init.d/openvpn stop
```

3 Complete step-by-step instructions

3.1 Kernel

Kernel support for TUN/TAP devices is necessary and there is a good chance it is already enabled. Check your running kernel's configuration for `CONFIG_TUN` setting. If it is `CONFIG_TUN=m`, then TUN/TAP driver module can be loaded by `modprobe tun`. In case of `CONFIG_TUN=y`, the driver is already included in the kernel itself. Otherwise you need to recompile the kernel with either `m` or `y`. Kernel configuration should be available at least as `/boot/config-x.y.z`. If there is more than one configuration in `/boot/`, check the one that corresponds to the kernel version reported by `uname -r`. The driver module should be loaded before running VPN. This can be done by adding `tun` as a separate line to `/etc/modules`:

```
# echo tun >> /etc/modules
```

3.2 Certificates and keys

OpenVPN at Math requires authentication using so-called Private Key Infrastructure (PKI). You will need:

1. Dartmouth Certificate Authority's (CertAuth1) own certificate;
2. Your personal certificate, signed by the CertAuth1;
3. Your private key, corresponding to your personal certificate.

There is a chance that certificates are already in your Netscape/Mozilla/Firefox. Go to ...Certificates → Manage Certificates. Then — Your Certificates (look for Dartmouth College and your name) and Authorities (look for Dartmouth):

- If there are no Your Certificates→Dartmouth College entries, go to https://collegeca.dartmouth.edu (DND-Based Enrollment) and get your Dartmouth certificate;
- If there is no Dartmouth College under Authorities, go and import it from https://collegeca.dartmouth.edu/rootcert.html.

Now we need to extract the certificates into separate disk files. Go again to ...Certificates → Manage Certificates → Your Certificates. Select (one of) your certificate(s) and click Backup. Enter `mydcpki`, note the location, Save. The newly created `mydcpki.p12` needs to be converted into PEM format. Use

```
$ openssl pkcs12 -nodes -in mydcpki.p12 -out mydcpki.pem
```

and enter the password you were asked to create while Backing up. The resulting `mydcpki.pem` should be a text file consisting of three `BEGIN`...`END` sections. Split the file accordingly:

```
-----BEGIN RSA PRIVATE KEY-----
```
(your private key; copy to `mydckey.pem`)
```
-----END RSA PRIVATE KEY-----
```

-----BEGIN CERTIFICATE-----
(Dartmouth CertAuth1 certificate; copy to collegeca.pem)
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
(your personal certificate signed by CertAuth1; copy to mydccert.pem)
-----END CERTIFICATE-----

Change access permissions on all the above files to 400 (i.e. read-only permission for owner only) for privacy and security.

N.B. The -nodes option tells openssl to export you private key in a form where the key is not protected by the pass phrase. This is convenient as you will not be asked for the pass phrase every time you start VPN. If you would rather like your private key to be pass-phrase-protected, do not include the -nodes option and openssl will prompt you to create your new pass phrase.

3.3 Install OpenVPN

```
# apt-get install openvpn
```

The same openvpn executable is used for both server and client. It's the configuration file that defines which mode, server or client, is used. Note that Debian installs openvpn to start automatically. This probably is not desirable for in case of a client. One way to disable it is to:

```
# mv /etc/rc2.d/S16openvpn /etc/rc2.d/s16openvpn
```

3.4 Configure

Get a ready client configuration file from gauss

```
# scp <you>@gauss:/usr/local/share/openvpn/mathvpn.conf /etc/openvpn/
```

or create /etc/openvpn/mathvpn.conf yourself with the following contents:

```
# Begin client config
tls-client
remote 129.170.28.64 1194
dev tun
proto udp
resolv-retry infinite
nobind
pull
ca /etc/ssl/certs/collegeca.pem
cert /etc/openvpn/mydccert.pem
key /etc/openvpn/mydckey.pem
cipher BF-CBC
tun-mtu 1500
fragment 1200
mssfix
mute-replay-warnings
ns-cert-type server
verb 4
```

```
log /etc/openvpn/client.mathvpn.log
# End of config
```

Move `mydc*.pem` files to `/etc/openvpn/` or adjust the config file accordingly. See Remarks at the end of this HOWTO if you are not the only person using this PC.

Move Dartmouth CertAuth1 certificate to `/etc/ssl/certs/` [1].

3.5 Test it

Check your current, i.e. non-VPN routing table. It may look like the one below, especially when on home network, behind a generic NAT firewall/router:

```
# route
Kernel IP routing table
Destination  Gateway Genmask        Flags Metric Ref Use Iface
192.168.0.0  *       255.255.255.0  U     0      0   0   eth0
default      gw      0.0.0.0        UG    0      0   0   eth0
```

Do a `traceroute` to some host on the Net, for example www.ams.org, and note the nearest routers.

Start OpenVPN client:

```
# /etc/init.d/openvpn start
```

Check `/etc/openvpn/client.log`. You may want to monitor your VPN log from a separate console:

```
# tail -f /etc/openvpn/client.mathvpn.log
```

Check the routing table again. It should look similar to this now:

```
# route
Kernel IP routing table
Destination            Gateway  Genmask         Flags Metric Ref Use Iface
hilbert.dartmouth.edu  gw       255.255.255.255 UGH   0      0   0   eth0
10.7.0.1               10.7.0.5 255.255.255.255 UGH   0      0   0   tun0
10.7.0.5               *        255.255.255.255 UH    0      0   0   tun0
192.168.0.0            *        255.255.255.0   U     0      0   0   eth0
default                10.7.0.5 0.0.0.0         UG    0      0   0   tun0
```

As you should see, there is now an additional network interface `tun0` on your system, which has been (dynamically) assigned 10.7.0.* IP numbers by the Math VPN server.

Run `traceroute`. You should see the following Dartmouth campus routers:

```
# traceroute www.ams.org
1 * * *
2 bradley.berry1-crt.dartmouth.edu (129.170.28.254) 64.641 ms ...
3 ropeferry-berry.ropeferry1-crt.dartmouth.edu (129.170.2.2) ...
4 core.border1-rt.dartmouth.edu (129.170.2.195) 71.783 ms ...
5 ... ... ...
```

[1] Alternatively, move the certificate to `/usr/share/ca-certificates/dartmouth/` and symlink to it from `/etc/ssl/certs/collegeca.pem`:
`# ln -s /usr/share/ca-certificates/dartmouth/collegeca.pem /etc/ssl/certs/collegeca.pem`
This is how Debian installs CA certificates.

3.6 Starting and stopping the client

Once your OpenVPN client has been setup and tested, switching it on and off is a matter of

```
# /etc/init.d/openvpn start
# /etc/init.d/openvpn stop
```

which of course is done as user root.

4 Remarks

Until now we silently suggested that there is only one OpenVPN configuration per workstation and the workstation is mostly used by one person, who is able to become root whenever needed. This will be true in many cases, but there can be situations where several OpenVPN configurations (for different OpenVPN servers and/or using different user credentials) have to coexist on one machine. For that purpose there may exist more than one configuration in `/etc/openvpn/`. The catch is that by default OpenVPN will try to create VPNs for all the `*.conf` files it finds in `/etc/openvpn/`. If this is not what you want, you can specify a particular VPN configuration on the command line. For example,

`# /etc/init.d/openvpn start` **`mathvpn`**

will start openvpn using `/etc/openvpn/`**`mathvpn`**`.conf` as configuration.

Also, user's security credentials—certificate and private key—do not have to be kept in `/etc/openvpn/`. If more than one person is using the machine, the best place for credentials is in user's home (`~/.tls/`, for example). Of course `key` and `cert` lines in `.conf` have to be adjusted accordingly.

References

[1] Hans-Cees Speel, Meet OpenVPN, published online in *Linux Journal*, December 15, 2004, http://www.linuxjournal.com/article/7949.

[2] OpenVPN home, http://openvpn.net.