目录

[一 介绍 1](#_Toc409598584)

[1.1 什么是Vpn 1](#_Toc409598585)

[1.2 OenVPN功能 1](#_Toc409598586)

[1.3 vpn实现原理. 1](#_Toc409598587)

[1.4 SSL VPN原理： 3](#_Toc409598588)

[二 安装 4](#_Toc409598589)

[1.1 安装环境 4](#_Toc409598590)

[2 软件包： 4](#_Toc409598591)

[3、编译安装 5](#_Toc409598592)

[4 、证书生成 6](#_Toc409598593)

[5 客户端配置 11](#_Toc409598594)

[6 添加路由（重要） 12](#_Toc409598595)

# 一 介绍

## 1.1 什么是Vpn

IP机制仿真出一个私有的广域网"是通过私有的隧道技术在公共数据网络上仿真一条点到点的专线技术。所谓虚拟，是指用户不再需要拥有实际的长途数据线路，而是使用Internet公众数据网络的长途数据线路。

1.2 OenVPN功能  
  
可以将两个不同的网段打通一条通道，可以互相访问，对于运维人员来说主要是为了安全，因为OpenVPNssl加密机制，增加了密码的安全度，运维人员可以自定义ip段进行管理服务器，简单点说，安全方便。

## 1.3 vpn实现原理.

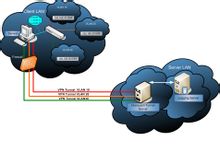
OpenVpn的技术核心是[虚拟网卡](http://baike.baidu.com/view/545.htm)，其次是[SSL协议](http://baike.baidu.com/view/254428.htm)实现，由于SSL协议在其它的词条中介绍的比较清楚了，这里重点对虚拟网卡及其在OpenVpn的中的工作机理进行介绍：

[虚拟网卡](http://baike.baidu.com/view/545.htm)是使用网络底层[编程技术](http://baike.baidu.com/view/146617.htm)实现的一个驱动[软件](http://baike.baidu.com/view/37.htm)，安装后在[主机](http://baike.baidu.com/view/23880.htm)上多出现一个网卡，可以像其它网卡一样进行配置。服务程序可以在[应用层](http://baike.baidu.com/view/239619.htm)打开[虚拟网卡](http://baike.baidu.com/view/545.htm)，如果[应用软件](http://baike.baidu.com/view/7886.htm)（如IE）向虚拟网卡发送数据，则服务程序可以读取到该数据，如果服务程序写合适的数据到虚拟网卡，应用软件也可以接收得到。[虚拟网卡](http://baike.baidu.com/view/545.htm)在很多的操作系统下都有相应的实现，这也是OpenVpn能够跨平台一个很重要的理由。

在OpenVpn中，如果用户访问一个远程的[虚拟地址](http://baike.baidu.com/view/1499823.htm)（属于[虚拟网卡](http://baike.baidu.com/view/545.htm)配用的地址系列，区别于真实地址），（TAP模则操作系统会通过路由机制将[数据包](http://baike.baidu.com/view/25880.htm)（TUN模式）或[数据帧](http://baike.baidu.com/view/1366167.htm)式）发送到虚拟网卡上，服务程序接收该数据并进行相应的处理后，通过SOCKET从外网上发送出去，[远程服务](http://baike.baidu.com/view/3297414.htm)程序通过SOCKET从外网上接收数据，并进行相应的处理后，发送给虚拟网卡，则[应用软件](http://baike.baidu.com/view/7886.htm)可以接收到，完成了一个单向传输的过程，反之亦然。

验证：

OpenVPN提供了多种[身份验证](http://baike.baidu.com/view/121510.htm)方式，用以确认参与连接双方的身份，包括：预享[私钥](http://baike.baidu.com/view/493846.htm)，



VPN在vlan中的应用

第三方证书以及用户名/密码组合。预享[密钥](http://baike.baidu.com/view/934.htm)最为简单，但同时它只能用于建立[点对点](http://baike.baidu.com/view/1145124.htm)的VPN；基于[PKI](http://baike.baidu.com/view/7615.htm)的第三方证书提供了最完善的功能，但是需要额外的精力去维护一个PKI证书体系。OpenVPN2.0后引入了用户名/口令组合的[身份验证](http://baike.baidu.com/view/121510.htm)方式，它可以省略[客户端](http://baike.baidu.com/view/930.htm)证书，但是仍有一份服务器证书需要被用作加密。

网路：

OpenVPN所有的[通信](http://baike.baidu.com/view/15007.htm)都基于一个单一的IP端口，默认且推荐使用UDP协议通讯，同时[TCP](http://baike.baidu.com/view/32754.htm)也被支持。OpenVPN连接能通过大多数的[代理服务器](http://baike.baidu.com/view/751.htm)，并且能够在[NAT](http://baike.baidu.com/view/16102.htm)的环境中很好地工作。[服务端](http://baike.baidu.com/view/1087294.htm)具有向[客户端](http://baike.baidu.com/view/930.htm)“推送”某些网络配置信息的功能，这些信息包括：IP地址、[路由](http://baike.baidu.com/view/18655.htm)设置等。OpenVPN提供了两种虚拟网络接口：通用[Tun/Tap](http://baike.baidu.com/view/8570754.htm)驱动，通过它们，可以建立三层IP隧道，或者虚拟二层以太网，后者可以传送任何类型的二层以太网络数据。传送的数据可通过[LZO](http://baike.baidu.com/view/5767680.htm)算法压缩。[IANA](http://baike.baidu.com/view/155085.htm)（Internet Assigned Numbers Authority）指定给OpenVPN的官方端口为1194。OpenVPN 2.0以后版本每个[进程](http://baike.baidu.com/view/19746.htm)可以同时管理数个并发的隧道。

OpenVPN使用通用网络协议（[TCP](http://baike.baidu.com/view/32754.htm)与[UDP](http://baike.baidu.com/view/30509.htm)）的特点使它成为[IPsec](http://baike.baidu.com/view/137005.htm)等协议的理想替代，尤其是在ISP（Internet service provider）过滤某些特定VPN协议的情况下。

在选择协议时候，需要注意2个加密隧道之间的网络状况，如有高延迟或者[丢包](http://baike.baidu.com/view/55894.htm)较多的情况下，请选择TCP协议作为底层协议，UDP协议由于存在无连接和[重传机制](http://baike.baidu.com/view/7142808.htm)，导致要隧道上层的协议进行重传，效率非常低下。

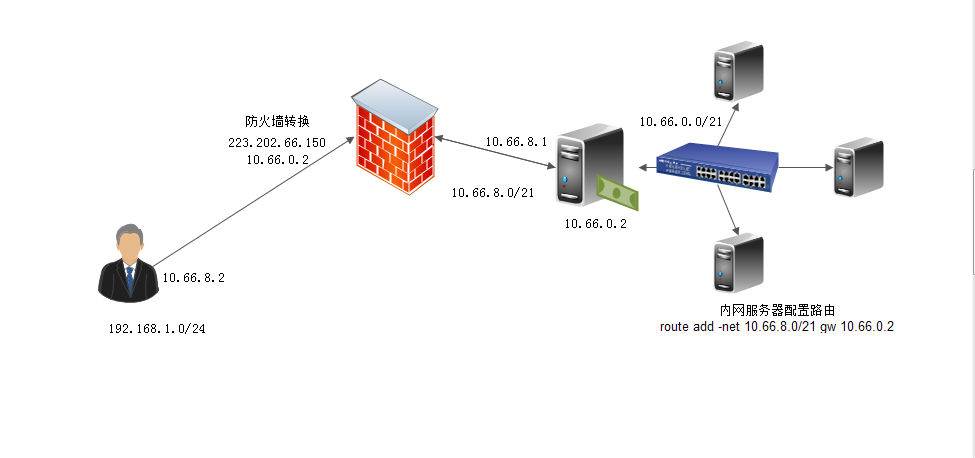
1.4 SSL VPN原理：  
 如果把SSL 和VPN 两个概念分开，大家对他们的含义应该都非常清楚，但是作为一种新技术，它们之间是如何结合起来的大家也许还不是很了解。从学术和商业的角度来讲，因为他们代表的含义有所不同，因而常常会被曲解。

SSL（安全套接层）协议是一种在Internet上保证发送信息安全的通用协议。它处于应用层。SSL用公钥加密通过SSL连接传输的数据来工作。 SSL协议指定了在应用程序协议（如HTTP、Telnet和FTP等）和 TCP/IP协议之间进行数据交换的安全机制，为TCP/IP连接提供数据加密、服务器认证以及可选的客户机认证。SSL协议包括握手协议、记录协议以及警告协议三部分。握手协议负责确定用于客户机和服务器之间的会话加密参数。记录协议用于交换应用数据。警告协议用于在发生错误时终止两个主机之间的会话。

VPN（虚拟专用网）则主要应用于虚拟连接网络，它可以确保数据的机密性并且具有一定的访问控制功能。VPN是一项非常实用的技术，它可以扩展企业的内部网络，允许企业的员工、客户以及合作伙伴利用Internet访问企业网，而成本远远低于传统的专线接入。过去，VPN 总是和IPSec 联系在一起，因为它是VPN 加密信息实际用到的协议。IPSec 运行于网络层，IPSec VPN 则多用于连接两个网络或点到点之间的连接。

所谓的SSL VPN，其实是VPN设备厂商为了与IPsec VPN区别所创造出来的名词，指的是使用者利用浏览器内建的Secure Socket Layer封包处理功能，用浏览器连回公司内部SSL VPN服务器，然后透过网络封包转向的方式，让使用者可以在远程计算机执行应用程序，读取公司内部服务器数据。它采用标准的安全套接层（SSL）对传输中的数据包进行加密，从而在应用层保护了数据的安全性。高质量的SSL VPN解决方案可保证企业进行安全的全局访问。在不断扩展的互联网Web站点之间、远程办公室、传统交易大厅和客户端间，SSL VPN克服了IPSec VPN的不足，用户可以轻松实现安全易用、无需客户端安装且配置简单的远程访问，从而降低用户的总成本并增加远程用户的工作效率。而同样在这些地方，设置传统的IPSec VPN非常困难，甚至是不可能的，这是由于必须更改网络地址转换（NAT）和防火墙设置。

通过SSL VPN远程访问企业内部网络的构架

网络规划

服务器内网 ：10.66.0.0/21 通过防火墙出去，网关防火墙10.66.7.254

Openvpn 服务器地址：10.66.0.2

Openvpn 隧道网段地址：10.66.8.0/21

客户端地址：192.168.1.0/24

# 二 安装

1.1 安装环境   
**环境：CentOS release 6.4 (Final) i386**

**协议：OpenVPN**

**端口：OpenVPN/TCP/1194(服务端), 1194为默认可修改；**

**加密：使用OpenSSL库来加密数据与控制信息，能够使用任何OpenSSL支持的算法。**

**认证：支持多种认证方式，包括预享私钥、第三方证书、用户名／密码组合；**

**服务：openvpn**

## 2 软件包：

**a、 openvpn-2.3.2.tar.gz**

**openvpn主程序，服务器和客户端均采用此包编译安装，不同的是两者证书及配置文件，其他环境一致。**

**b、 easy-rsa-2.2.2.tar.gz**

**This is a small RSA key management package,based on the openssl command line tool, thatcan be found in the easy-rsa subdirectory of the OpenVPN distribution. ——RSA证书生成管理软件**

**c、pam CentOS6x/RHEL6x自带无需安装,yum更新便可**

**PAM (Pluggable Authentication Modules)是由Sun提出的一种认证机制。它通过提供一些动态链接库和一套统一的API，将系统提供的服务和该服务的认证方式分开，使得系统管理员可以灵活地根据需要给不同的服务配置不同的认证方式而无需更改服务程序。PAM最初是集成在Solaris中，目前已移植到其它系统中，如Linux、SunOS、HP-UX 9.0等。**

**c、lzo (可yum安装)**

**The OpenVPN RPM package requires the LZO library for real-time link compression. ——用于实时数据压缩**

**e、openssl (系统自带，可yum更新)**

**SSL，Security Socket Layer,是一个安全传输协议，在Internet网上进行数据保护和身份确认。OpenSSL是一个开放源代码的实现了SSL及相关加密技术的软件包，由加拿大的Eric Yang等发起编写。**

**软件下载：**

**http://swupdate.openvpn.org/community/releases/openvpn-2.3.2.tar.gz**

**https://codeload.github.com/OpenVPN/easy-rsa/tar.gz/2.2.2**

**http://swupdate.openvpn.org/community/releases/openvpn-install-2.3.2-I003-i686.exe**

## 3、编译安装

**注：服务器端环境和客户端环境是一样的，只是配置文件不同而已一个是server.conf另一个是client.conf，另外证书文件也是不同的。**

**3.1.安装依赖包**

**# yum -y install pam pam-devel**

**# yum -y install openssl openssl-devel**

**# yum -y install lzo lzo-devel**

**把系统的Forward打开，可以通过如下命令实现**  
**echo 1 > /proc/sys/net/ipv4/ip\_forward  
或者**

**使用sysctl -w net.ipv4.ip\_forward=1  
或者**

**修改/etc/sysctl.conf文件，增加  
net.ipv4.ip\_forward = 1**

**3.2.编译openvpn**

**# tar xf openvpn-2.3.2.tar.gz**

**# cd openvpn-2.3.2**

**# ./configure -prefix=/usr/local/openvpn**

**# make && make install**

**# echo $?**

**目录规划（可自定义）**

**# mkdir /usr/local/openvpn/conf #配置文件位置，如server.conf**

**# mkdir /usr/local/openvpn/keys #证书文件位置**

**（安装到/usr/local/openvpn方便后期管理**

## 4 、证书生成

**4.1.环境配置**

**# tar xf easy-rsa-2.2.2.tar.gz**

**# cd easy-rsa-2.2.2**

**# cp -R easy-rsa/ /usr/local/openvpn/**

**# cd /usr/local/openvpn//easy-rsa/2.0**

**# vi vars #设置变量**

**export KEY\_COUNTRY="CN" #(国家)**

**export KEY\_PROVINCE="BJ" #（省份）**

**export KEY\_CITY="BJ" #（城市）**

**export KEY\_ORG="IT" #（组织）**

**export KEY\_EMAIL=**[**me@myhost.mydomain**](mailto:me@myhost.mydomain) **#（邮件地址）**

**export KEY\_OU="huirendai" #（单位）**

**注：其他不作修改，默认便可**

**# source ./vars #使变量生效**

**# ./clean-all #初始化keys目录，创建所需要的文件和目录**

**# ln -s openssl-1.0.0.cnf openssl.cnf #做SSL配置文件软链：**

**上面是初始化工作，以后在进行证书制作工作时，仍旧需要进行初始化，但只需要进入openvpn\easy-rsa目录，运行vars就可以了，不需要上面那些步骤了。**

**4.2 证书生成**

**# ./build-ca #生成ROOT CA证书，用于签发Server和Client证书（ca.key ca.crt）**

**# ./build-dh #生成公开密钥算法（dirrie-hellman）文件 （dh2048.pem）**

**# ./build-key-server server #生成服务器证书和密钥（server.key server.crt server.csr 01.pem）**

**# ./build-key client #生成客户端证书和密钥（client.key client.crt client.csr 02.pem）**

**注：生成服务器和客户端证书时提示输入额外属性，此处可不填！**

**Please enter the following ‘extra’ attributes**

**to be sent with your certificate request**

**A challenge password []:**

**An optional company name []:**

**Certificate is to be certified until Jul 16 11:16:39 2023 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**

**其中有两个选择的地方，都选择y即可，如果报错comaname的错误时是你的输入的名字错误，或是重名了。修改文件名即可。  
4.3.拷贝证书到相应位置**

**# cd /usr/local/openvpn/easy-rsa/2.0/**

**cp keys/\* /usr/local/openvpn/keys/**

**4.4 拷贝配置文件到相应位置  
cp /opt/software/openvpn-2.3.2/sample/sample-config-files/server.conf /usr/local/openvpn/conf/**

**4.5 建立客户端证书和密钥**

**如果以后要为其他Client颁发证书，直接使用build-key命令签发新证书。  
# ./build-key client #client 为你自己起的客户端证书名字  
Generating a 1024 bit RSA private key  
........++++++  
....................++++++  
writing new private key to 'elm.key'  
-----  
。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。**

**。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。（略）**

**Name(eg, your name or your server's hostname) []:client1 #注意Common Name最好不要相同，如果相同[server要加duplicate-cn选项，那么Email地址也不能相同  
Email Address [**[**feiyang@163.com**](mailto:feiyang@163.com)**]:**

**……………….**

**…………………(略，回车即可)**

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

**说明创建成功**

**如果想生成多个client key的话，重复以上步骤，修改client1为client2，client3……即可。**

**4.6 配置服务文件**

**local 10.66.0.2**

**port 1194**

**proto tcp**

**dev tun**

**ca /usr/local/openvpn/keys/ca.crt**

**cert /usr/local/openvpn/keys/server.crt**

**dh /usr/local/openvpn/keys/dh2048.pem**

**server 10.66.8.0 255.255.248.0**

**ifconfig-pool-persist ipp.txt**

**push "route 10.66.0.0 255.255.248.0"**

**push "dhcp-option DNS 10.66.0.2"**

**keepalive 10 120**

**comp-lzo**

**max-clients 100**

**persist-key**

**persist-tun**

**status openvpn-status.log**

**log-append /var/log/openvpn.log**

**verb 3**

**4. 7 配置文件解释**

**；local a.b.c.d #申明本机使用的IP地址，本地的那个IP 地址被openvpn监听  
 port 1194 #申明使用的端口，默认1194，可以修改**

**如果采用http proxy 必须使用tcp 协议  
proto tcp #申明使用的协议，默认使用UDP，如果使用HTTP proxy，必须使用TCP协议（两种协议选一个 一般是udp）  
 ；proto udp**

**dev tap #申明使用的设备可选tap和tun，tap是二层设备，支持链路层协议。**

**将创建一个以太网隧道**

**；dev tun #tun是ip层的点对点协议，限制稍微多一些，本人习惯使用TAP设备**

**（两种都可以用选择一种）**

**将创建一个路由隧道**

**；dev-node MyTap**

**ca /usr/local/openvpn/keys//ca.crt #OpenVPN使用的ROOT CA，使用build-ca生成的，用于验证客户是证书是否合法 (这里的路径一定要正确，是你刚才生产证书的路径)**

**cert /usr/local/openvpn/keys/server.crt #Server使用的证书文件**

**key /usr/local/openvpn/keys/server.key #This file should be kept secret #Server使用的证书对应的key，注意文件的权限，防止被盗**

**crl-verify /usr/local/openvpn/keys/vpncrl.pem #CRL文件的申明，被吊销的证书链，这些证书将无法登录**

**dh /usr/local/openvpn/keys/dh2048.pem #上面提到的生成的Diffie-Hellman文件**

**；server 10.8.0.0 255.255.255.0 #用于记录某个Client获得的IP地址，类似于dhcpd.lease文件 # 配置VPN使用的网段，OpenVPN会自动提供基于该网段的DHCP 服务，但不能和任何一方的局域网段重复，保证唯一 。server端ip默认会设为.1的地址。**

**维护客户端和虚拟ip地址联系，在openvpn 重启是，再次连接的客户端将依然被分配和以前一样的ip**

**server 10.66.8.0 255.255.248.0**

**ifconfig-pool-persist ipp.txt #防止openvpn重新启动后“忘记”Client曾经使用过的IP地址**

**；server-bridge10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100 #Bridge状态下类似DHCPD的配置，为客户分配地址，由于这里工作在路由模式，所以不使用**

**如果配置这个选项，会导致链接到VPN以后，走VPN的网关不能上网**

**；push "redirect-gateway def1 bypass-dhcp"**

**下面的配置可以使客户端网络能访问服务器后面的网络的服务器**

**如果后面的服务器的网络是10.66.0.0 段的需要加入下面文件**

**；push"route 192.168.10.0 255.255.255.0" #通过VPN Server往Client push路由，client通过pull指令获得Server push的所有选项并应用(这一点很重要，这样其他服务器添加一下路由就可以登陆了，否则无效，这个最好和你的内网断一至，否则也不生效)  
push"route 192.168.1.0 255.255.255.0"**

**；push"route 192.168.20.0 255.255.255.0"**

**如果需要服务器后面网络访问客户端网络假如客户端网络地址（192.168.40.0段）需要：**

**client-config-dirccd**

**route192.168.40.0 255.255.255.0**

**然后在服务器端ccd 目录下创建一个文件，文件名是客户端的公共名**

**文件内容：**

**iroute 192.168.40.0 255.255.255.0**

**；route192.168.99.0 255.255.255.0**

**；client-config-dirccd**

**；route10.9.0.0 255.255.255.252**

**；learn-address./script**

**；push"redirect-gateway"**

**#如果可以让VPN Client之间相互访问直接通过openvpn程序转发，**

**；push"dhcp-option DNS 10.8.0.1"**

**；push"dhcp-option WINS 10.8.0.1"**

**#不用发送到tun或者tap设备后重新转发，优化Client to Client的访问效率  
；client-to-client**

**；duplicate-cn**

**Keepalive 10 120 #NAT后面使用VPN，如果VPN长时间不通信，NAT Session可能会失效，导致VPN连接丢失，为防止之类事情的发生，keepalive提供一个类似于ping的机制，下面表示每10秒通过VPN的Control通道ping对方，如果连续120秒无法ping通，认为连接丢失，并重新启动VPN，重新连接（对于mode server模式下的openvpn不会重新连接）。**

**没有HMAC signature的控制信息不予处理，注意server端后面的数字肯定使用0，client使用1**

**；tls-auth ta.key 0 # This file is secret**

**comp-lzo #对数据进行压缩，注意Server和Client一致**

**max-clients 100 #定义最大连接数**

**;User nobody #定义运行openvpn的用户**

**;group nobody**

**persist-key #通过keepalive检测超时后，重新启动VPN，不重新读取keys，保留第一次使用的keys**

**persist-tun #通过keepalive检测超时后，重新启动VPN，一直保持tun或者tap设备是linkup的，否则网络连接会先linkdown然后linkup**

**Status openvpn-status.log #定期把openvpn的一些状态信息写到文件中，以便自己写程序计费或者进行其他操作**

**Log openvpn.log #记录日志，每次重新启动openvpn后删除原有的log信息**

**;log-append openvpn.log #记录日志，每次重新启动openvpn后追加日**

**Verb 3 #相当于debug level，定义上面日志级别，具体查看manual。**

**;mute20**

**4.8 启动服务**

**/usr/local/openvpn/sbin/openvpn –daemon --config /usr/local/openvpn/conf/server.conf**

## 5 客户端配置

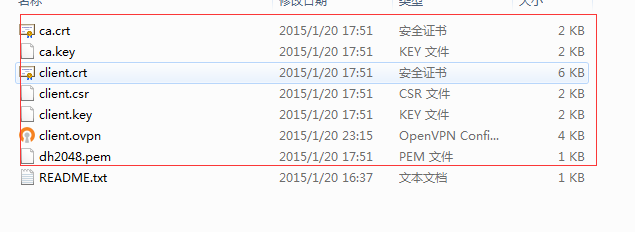
**5.1 配置windows openvpn 客户端**

**安装windows客户端**

**软件下载地址：**[**http://swupdate.openvpn.net/community/releases/openvpn-2.2-beta3-install.exe**](http://swupdate.openvpn.net/community/releases/openvpn-2.2-beta3-install.exe)

**下载安装，没得说，widows 下一步下一步**

**获取服务器生成的证书  
将服务器端生成的client.crt , client.key 还有ca.crt 下载到本地,放到openvpn的安装目录的config目录下,如我安装的路径为**

**D:\Program Files (x86)\OpenVPN\config ,此目录需要放server端生成证书文件：**

**把安装目录D:\Program Files (x86)\OpenVPN\sample-config\Client.ovpn**

**拷贝到D:\Program Files (x86)\OpenVPN\config**

**下面是客户端配置文件client.ovpn的详解，这个配置文件需要和服务端相对应**

**client**

**dev tun**

**proto tcp**

**remote 223.202.66.150 1194**

**resolv-retry infinite**

**nobind**

**persist-key**

**persist-tun**

**ca ca.crt**

**cert client.crt**

**key client.key**

**ns-cert-type server**

**comp-lzo**

**verb 3**

**配置文件详解**

**；client**

**dev tun**

**；dev tap #详解看上面server.conf**

**；dev-node MyTap**

**proto udp**

**；proto tcp #详解看上面server.conf**

**Remote serverip1 port**

**；Remote serverip2 port #你的vpn的公网ip，就是用来连接服务端的,port服务端是什么端口,这里就是什么端口**

**；remote-random #如果有两个服务器地址，这个选项就是轮询的意思**

**resolv-retry infinite**

**nobind**

**;user nobody**

**;group nobody**

**persist-key**

**persist-tun**

**；http-proxy-retry # retry on connection failures**

**；http-proxy [proxy server] [proxy port #]**

**；mute-replay-warnings**

**ca ca.crt**

**cert client1.crt**

**key client1.key #此处不像server.conf 需要写绝对路径 ）**

**ns-cert-type server**

**；cipher x**

**；tls-auth ta.key 1**

**comp-lzo**

**verb 3**

**# Silence repeating messages**

**;mute 20**

## 6 添加路由（重要）

**在内网服务器上添加路由，这一项很重要，否则openvpn不能直接访问。**

**route add -net 10.66.8.0/21 gw 10.66.0.2 (openvpn 服务器的地址为下一条地址)**

**(需要在内网的服务器上添加vpn 隧道段的回路路由，这个很重要 )**