这本指南来自 psaroskalazines.gr
出版的一系列 VPN 小册子

插图由作者手绘
排版设计 scribus 软件完成

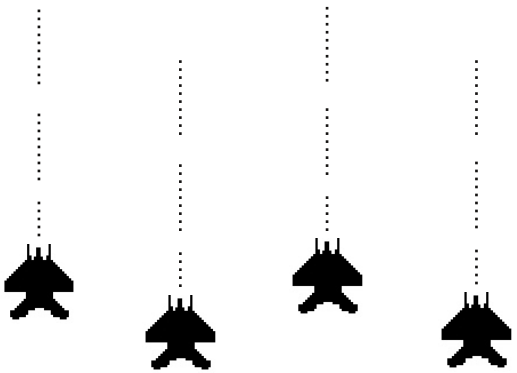
字体：
方正像素 24

致谢：
digitaldefenders.org 的资金资助
systerserver.net 为小册子提供
git 托管服务

在 mastodon 长毛象上联系作者：
systerserver.town/@mara



# OpenVPN 故障排除



如何 OPENVPN
安装和维护个人托管的 VPN 服务

# 速查表

安装指南：
openvpn.net/community-resources/installing-openvpn/
community.openvpn.net/openvpn/wiki/HOWTO#InstallingOpenVPN
还有 https://openvpn.net/community-downloads/
和 常见问题 https://openvpn.net/community-resources/

吊销证书：
openvpn.net/community-=resources/revoking-certificates/

把 openvpn 作为服务运行：
https://www.smarthomebeginner.com/configure-openvpn-to-autostart-linux/

julia evan' 关于网络排障的相关小册子：
https://jvns.ca/tcpdump-zine.pdf

如何使用 openvpn 把所有网络交通重新定向：
https://openvpn.net/community-resources/how-to/#redirect

如何把一个 debian 服务器为了所有的网络交通，单独作为 openvpn 服务来运行，包括的需要的 ip 规则：
www.linode.com/docs/networking/vpn/tunnel-your-internet-traffic-through-an-openvpn-server/

VPN 是什么，怎么运作：
https://archive.org/details/vpn-zines

# 建立路由规则

如果我们要通过 openvpn 来访问受限制的网站，我们需要在 openvpn 服务器的路由表里添加如下规则：

```
<cmd>
# Allow traffic on the tunnel (tun0) interface.
-A INPUT -i tun0 -j ACCEPT
iptables -A FORWARD -i tun0 -j ACCEPT
iptables -A OUTPUT -o tun0 -j ACCEPT
# Allow forwarding traffic only from the openVPN.
iptables -A FORWARD -i tun0 -o eth0 -s 10.4.0.0/24 -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t nat -A POSTROUTING -s 10.4.0.0/24 -o eth0 -j MASQUERADE
# Install iptables-persistent for keeping the rules across reboots
sudo apt install iptables-persistent
dpkg-reconfigure iptables-persistent
# Forward IPv4 traffic
echo 'net.ipv4.ip_forward=1' >> /etc/sysctl.d/99-sysctl.conf
sysctl -p # activates the above change
systemctl restart openvpn-server@server.service
</cmd>
```

然后我们把 client.conf 里的 "client" 项换成 "tls-client"，接着重启 openvpn。

如果我们不能成功远程连接，我们可以用 traceroute 查看可能的错误：

```
<cmd> traceroute -i tun0 youtube.com </cmd>
```

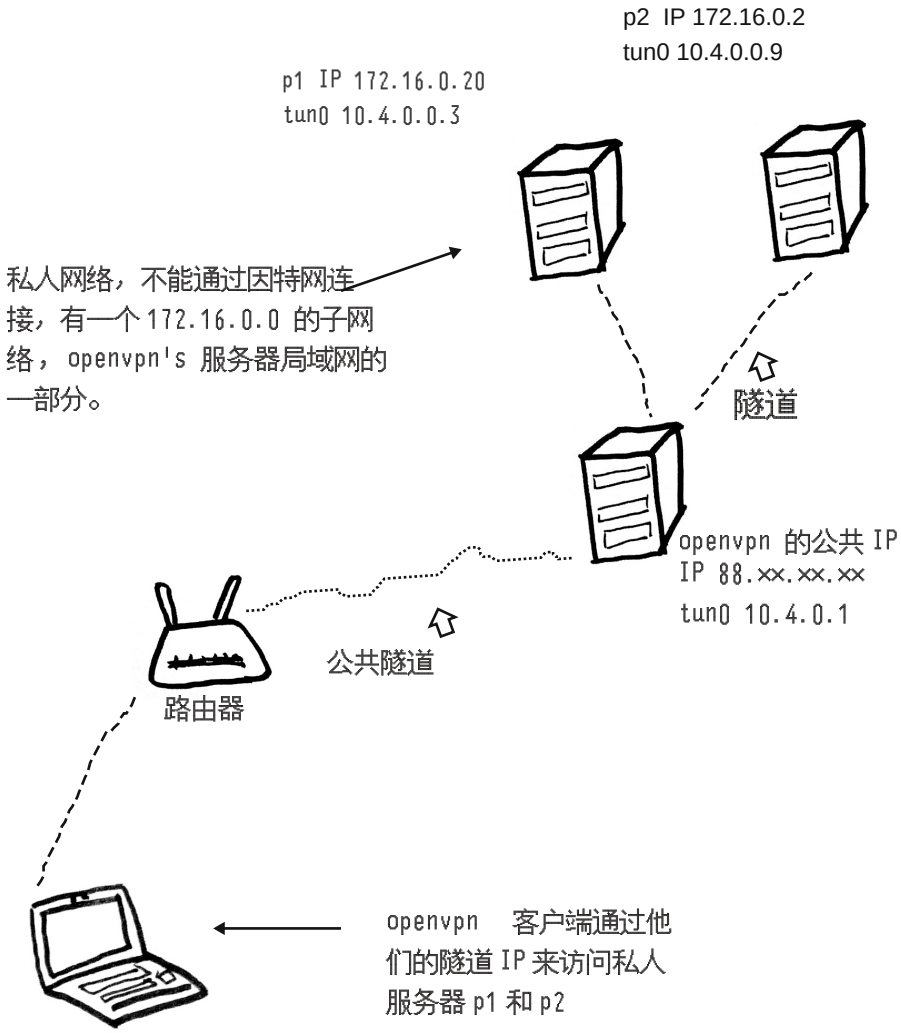
当上述命令在运行时，我们尝试在浏览器内访问 youtube，traceroute 可以显示连接在哪里停止了。有可能是服务器或客户端的错误，这样我们可以回到之前相应的故障排除步骤。



# 简介

这本小册子指引读者来进行 openVPN 的安装和故障排除环节。这是面向所有想在个人维护，自治的服务器里运行 vpn 服务的系统管理员的一本指南，不论 ta 们的能力级别。这本指南也面向想网络通讯、隐私和网络安全对进行试验的用户。册子的内容涉及了使用终端和命令行，也就是 Command Line Interface - CLI，并在 Linux 系统环境下操作。作者使用的语言是简单易懂的；曾经有着 26 岁（稍晚）才接触网络通讯的经历，作者深知在竞争激烈的电脑极客和黑客中作为新鲜人的感受。

# 私人和公共网络


p2 IP 172.16.0.2
tun0 10.4.0.0.9
p1 IP 172.16.0.20
tun0 10.4.0.0.3
私人网络，不能通过因特网连接，有一个 172.16.0.0 的子网络，openvpn's 服务器局域网的一部分。
隧道
openvpn 的公共 IP
IP 88.xx.xx.xx
tun0 10.4.0.1
公共隧道
路由器
openvpn 客户端通过他们的隧道 IP 来访问私人服务器 p1 和 p2

# 访问受限制的网站

当 openvpn 在服务器和客户端上都开始运行后，我们就可以访问 openvpn 服务器后私人的服务器，或者访问受限制的网站了。我们也可以把我们所有的网络交通都通过 openvpn 来运输，但是如果我们用户的电脑是使用动态 IP 的话，这样做会使到是我们的网络交通变得非常慢而且不稳定。我们可以看速查表来查找相关资料。

在这里，我们可以看一下如何访问在地理上被限制的网站，或者是通过 openvpn 的局域网络来访问一个私人的服务器。我们需要把我么想访问的域名、私人 IP 地址、或者子网络地址加入到 server.conf 文件里。

比如说如果在我们住的地方，youtube 的访问是受限的话，我们可以在 server.conf 里添加下列信息：

```
push "route youtube.com 255.255.255.255"
```

如果我们想连接到一个私人的服务器，我们可以添加它的 IP 地址或者是它的私密子网络：

```
push "172.16.0.0 255.255.0.0"
```

提示 1：在这个例子中，openvpn 服务器也是一个网关。
提示 2：我们基本上把私人的子网络加到了 openvpn 的服务器上。



# 目录

# 清单

服务器： openVPN 在服务器模式下运行的机器，并将来自用户的网络交通传送到因特网或者是私人网络，可以是虚拟的私人服务器，或者是实体的金属服务器。

客户端： 用户用来使用 openVPN 来连接因特网的设备，比如说电脑或者是智能手机。

WAN 和 LAN: 广域／局域网，局域网又被称为私人网络，是不能通过因特网访问的。

路由器： 把一个网络的数据发送到另一个网络的机器。它有内置的以太网端口，家用的路由器通常有一个 Wi-Fi 接入点和内置的调制解调器／猫，来用电缆或者数字用户线路来输送网络交通。由路由器传输的网络交通使用 TCP/IP 协议格式，并且路由器会控制什么交通可以通过。

网关： 比上面的定义更宽泛的一个定义，指代所有能够和因特网互相发送和收取数据包的机器。可以是一个电缆或者数字用户线路的猫，一个路由器，或者是设置为防火墙的服务器。

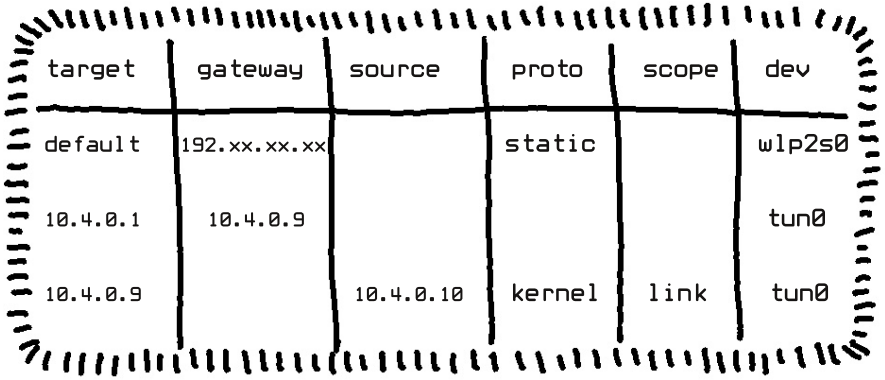
pc: 个人电脑

IANA: 也就是互联网号码分配局， Internet Assigned Numbers Authority，管理全球的 IP 地址分配。



# 路由表

＊使用 "route" 或者 "routel" 命令在客户端的电脑上查看路由表。查看服务器的隧道 IP 是否在目标栏（target/destination）中，客户端的隧道 IP 是否在源（source）栏中。例如，如果客户端的隧道 IP 是 10.4.0.10， 服务器的隧道 IP 是 10.4.0.1，那么表应该看起来像这样：




加入我们在隧道的另一端连接了别的服务器或者客户端的话，它会显示为一个连接，并有一个 tun0 IP，比如 10.4.0.9。在下一章我们来看如何添加连接（一个私人的服务器或者是受限制的网站）。

# 客户端排障

* 在 client.conf 文件里查找 "verbosity" 并增加到 6 或者更多，重新运行 openvpn。不要忘记当 vpn 连接修好后，把它调回 3 或更低，不然记录文件会占用太多的空间。

* 接着，不通过服务，直接运行 openvpn，检查终端输出的记录看有什么错误：
<cmd>sudo openvpn --config client.conf </cmd>
通过 "ctr+C" 停止输出。

*需要确认，在相关联的项中，客户端的设定和服务器的设定是一致的。 终端显示的错误能够指出哪些配置需要更正。

* ping 服务器的隧道 IP 来看是否能被客户端的电脑连接的到。这个可以从我们在 server.conf 中设定的的隧道子网络得出，服务器占用第一个地址。



# 命令行

ifconfig：显示我们设备的 IP 地址
ip addr：ifconfig 的新一代
nslookup：显示域名的 IP 地址
traceroute：对从一个 IP 网络传输到指定目的地的数据包进行追踪。
EOF：文件的最后一行
echo "some text" >> filename：在文件的最后一行添加 sometext 字段
cat filename1 >> filename2：在文件最后一行添加内容
routel：用可读的格式罗列设备的路径
ping：检查和服务器的连接
例子：ping systerserver.net 命令的输出：
64 bytes from systerserver.net (xx.xx.xx.xx): icmp_seq=1 ttl=47 time=97.3 ms
64 bytes from systerserver.net (xx.xx.xx.xx): icmp_seq=2 ttl=47 time=95.6 ms
这个表明此域名是可以被连接的。
tail or less：用来阅读记录，例如
sudo tail -n 100 /var/log/syslog，option -n 100
显示文件的最后 100 行
vim，emacs 或者 nano：用来编辑配置文件的文本编辑程序，最后一个比较直观易懂。
提示：在一个只能通过 SSH 来连接的远程服务器上，我们需要使用上述之一的终端行用的文本编辑器。在个人电脑上我们可以使用，不包含随机符号的文本编辑器。比如说在一些 windows 的文本编辑器里，^M 符号会在一行输入的最后被添加。
<cmd>，</cmd>：命令行块
#：对命令行的注释

# 下载和安装

1. openvpn（2. x版本）
2. easy-rsa（如果没有随openvpn安装的话）

有一些linux的分支需要安装openssl，lzo和pam的依赖。对于其他的操作系统，可以参考openvpn线上的文档。（参考册子的速查表）.

根据我们系统的不同，我们可以使用yum，apt-get，或者 emerge安装。
这里我们使用一个debian/ubuntu的例子
<cmd> sudo apt install openvpn </cmd>

提示1：在OpenVPN 配置中，我们需要选择一个被路由或者被桥接的网络。在大多数情况下，当我们需要安全的访问网络或者规避审查机制时，我们需要一个被路由的VPN，而不是被桥接的。下一页的表会对两者的不同作出说明。

提示2：接着，我们需要选择一个私人的子网络来作为openvpn的隧道。IANA提供的可用的子网络包括：
10. 0. 0.0/8，172.16. 0.0/12，192.168. 0.0/16. 在这本指南里我们使用第一个。



# 服务器端排障

* 检查服务器的openvpn是否在运作：
<cmd> sudo systemctl status openvpn-server@server </cmd>
输出的消息将提示有什么错误。我们可以用左右箭头来查看消息。
如果没有在运行的话，我们可以用这行命令来启动运行：
<cmd> sudo systemctl start openvpn-server@server </cmd>

* 打开记录文件，查看错误消息（记录文件的路径在server.conf里）：
<cmd> tail-n 100 /var/log/openvpn </cmd>
如果记录文件在别处的话，我们需要把文件路径换成正确的路径。
如果你想阅读文件，同时保持将新消息添加到文件里的话：
<cmd> less +F /var/log/openvpn </cmd>
如果你想阅读文件，但是停止添加新消息的话，我们可以输入"ctr + C"。
输入大写的F，我们可以在此在文件末端添加新消息。通过"ctr+C" 然后 "q"来离开文件。

* 查看隧道IP是否存在：
<cmd> sudo ifconfig tun0 #for Ubuntu </cmd>
<cmd> sudo ip add show dev tun0 #Debian </cmd>

* ping 客户端的隧道IP，可以使用如下命令在客户端设备查看：
ifconfig or ip addr

# 运行服务

当 openvpn 安装好后，我们可以在 linux 服务器和客户端上重启它，通过编辑 /etc/default/openvpn， 把 "AUTOSTART=all" 前面的 "#" 移除。
对于系统单元，我们可以输入以下命令：

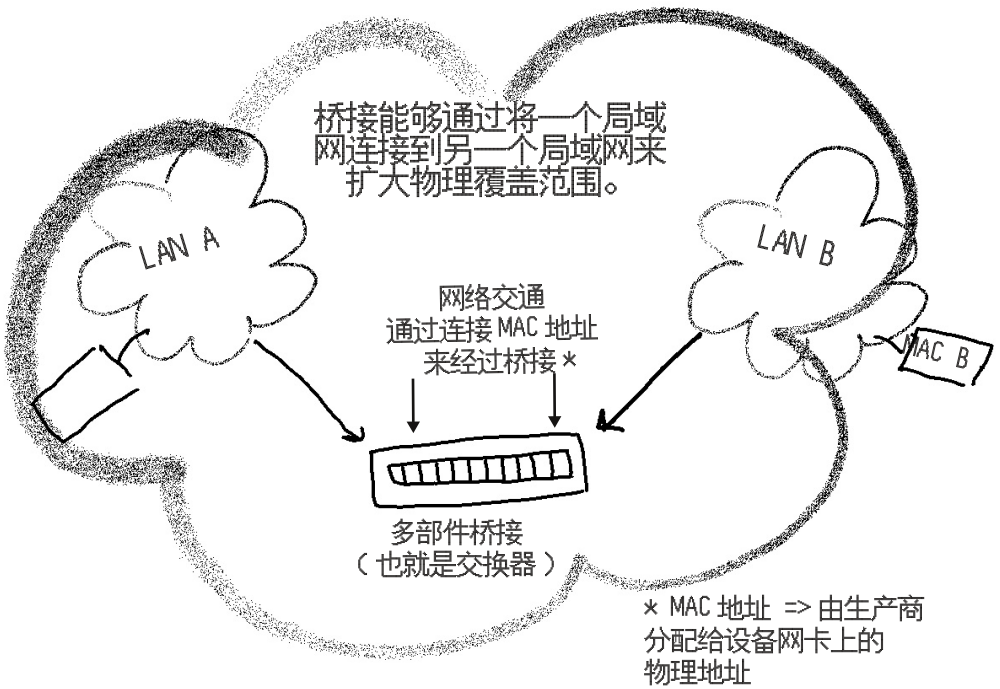
```
<cmd>
sudo systemctl enable openvpn-server@<name>.service
sudo systemctl daemon-reload
sudo systemctl restart openvpn-server@<name>.service
#for the client machine it's openvpn-client@<conf>.service
</cmd>
```

说明：
openvpn 的系统单元源码位于 /lib/systemd/system/。 比如说，如果要在客户端运行 openvpn 的话，系统单元源码就是 "openvpn-client@.service"。
我们在 "openvpn-client@" 和 ".service" 之间输入的字节，比如说 <name>，会接着输入进系统单元源码中，来运行下列命令：
进入 /etc/openvpn 目录并运行 "openvpn --config <name>.conf".
所以，把 client.conf 文件放到 /etc/openvpn 目录下是良好的操作规范；或者在源码中，把 client.conf 的路径换掉。（使用超级用户权限）。

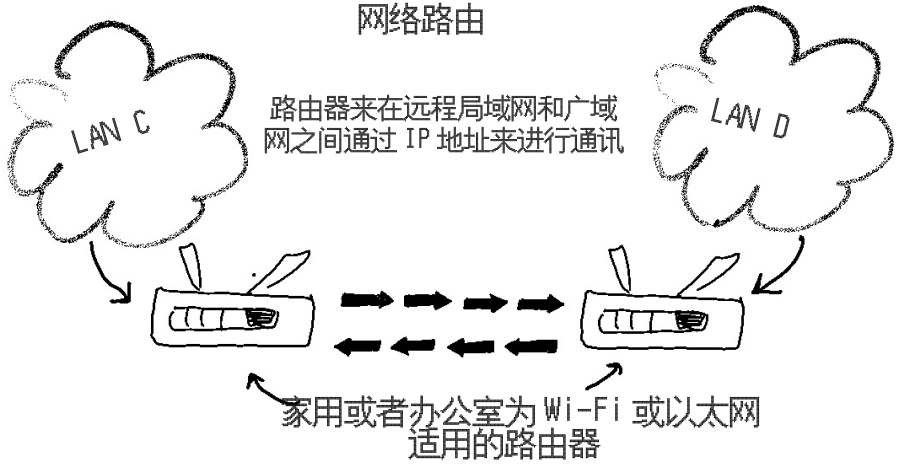
可以吗？如果不行的话，我们来看如何排除故障：
常见的问题服务器和客户之间的连接不能被建立。



网络桥接




网络路由



# 生成证书

我们在 easy-rsa 里会看到许多目录的文件！ 我们来开始编辑 vars 文件来设置变量，这样我们可以为接下来生成证书节省时间。

<cmd> vi vars #or nano </cmd>

"vars" 文件是跟证书颁发机构有关的，这个机构会分发和认证服务器和客户端的证书。所哟我们使用和我们团队或者地址有关的名字。我们也将 "KEY-SIZE = 2048" 设置到更高，比如说 4096。当我们保存并关闭 var 文件后，我们需要更新变量，这样终端能够记住我们刚刚配置的数值：

cmd> . ./vars </cmd>

删除旧的或不用的证书

<cmd> ./clean-all </cmd>

提示：上面的命令会删除所有现有的钥匙。

我们在设置一个 X 509 PKI 的 VPN （也就是 Public Key Infrastructure，公开密钥基础设施，看下一页的解释。这个用来建立使用证书和钥匙的认证方法。easy-rsa 软件可以帮我们生成这些文件。

如果它随着 openvpn 一起装好的话，那么可以在 "/etc/openvpn/easy-rsa" 被找到。找不到吗？ 使用一下命令安装：

<cmd>
sudo apt install easy-rsa
locate easy-rsa | grep README
less <path/to/README>
</cmd>

README 文件解释了如何生成指向 easy-rsa 源代码的符号链接。这样我们可以在 openvpn 里使用这些源代码。



# 客户端配置

客户端配置文件的路径和服务器的配置文件路径相似，

包括了证书颁发机构文件的绝对或相对路径（ca.crt），客户端的钥匙，还有为 HMAC 额外生成的钥匙，里面包含了 "tls-auth ta.key 1" 项。

我们还需要添加 VPN 服务器的远程 IP 或者域名地址，还有我们在 server.conf 中配置的端口：

remote <public ip> 1196

上述的四个文件需要发给使用 VPN 服务的用户。这些文件可以用标签的方法来在 client.conf 中打包。

如果采取这个方法的话，我们需要注释掉这些文件在 client.conf 的路径：

; ca ca.crt
; cert client.crt
; key client.key

我们换成：

<ca>
CA 证书
</ca>
<cert>
Client's 证书
</cert>
<key>
客户端钥匙
</key>
<tls-auth>
HMAC 钥匙
</tls-auth>

# 配置小提示

如果我们要提供用户组的证书（同样包括多个用户的情况），我们取消"duplicate-cn"的注释．我们也可以对 keepalive 项取消注释。

给密钥散列消息认证码，也就是 HMAC ，生成额外的钥匙是一个好的规范，因为它在 server.conf 文件中被提及了。

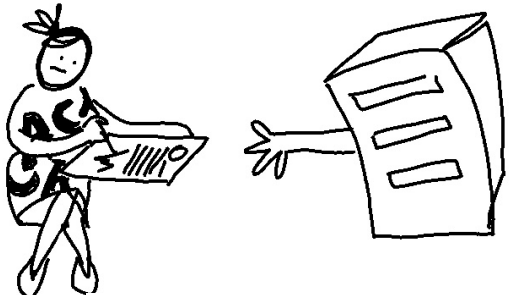
```
tls-auth ta.key 0
mode server
tls-server
```

对于加密层，较好的选择有：
cipher AES-256-CBC，auth SHA256，和 key-direction 1.

我们给记录的文档添加一个路径，最后把“详细模式”的配置成“verb 3”，如果我们以后需要排除故障的话我们可以以后再看。证书吊销的选项将在取消用户访问权限时会非常有用。我们可以在 server.conf 文件的最后来配置这个选项：

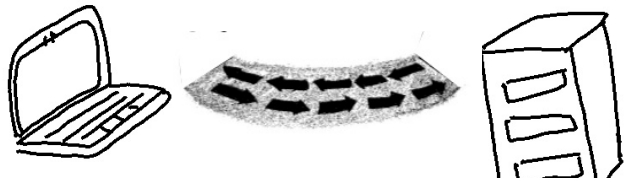
```
crl-verify crl.pem
See link reference at CHEAT SHEET.
```





证书颁发机构为服务器和客户端的证书签名

客户端收到由 openvpn 服务器管理员发来的证书，钥匙 和由证书颁发机构办法的证书



客户端和服务器将使用 tls 认证和密钥交换协议来协商隧道连接

# 证书颁发机构，服务器，和客户端密钥

为证书颁发机构创建钥匙（也就是 CA）：
<cmd> . /build-ca </cmd>
对出现的问题输入 "enter"，最后两个问题输入 "yes"。
在最后，检查钥匙有没有在目录 "/etc/openvpn/easy-rsa/keys" 里。

为服务器生成钥匙：
<cmd> . /build-key-server <name-of-our-server> </cmd>
对大多数的问题都输入 "enter"，除了以下问题：
"common NAME" for 这里我们可以输入服务器的名字或者我们团队／机构的名字。
这最后两个问题我们输入 "y"：
1."Sign the certificate?" （对证书签字吗）
2. "1 out of 1 certificate-requests certified, commit?"
（证书的提交已经被认证，是否提交？）

为用户，也就是客户端生成钥匙：
<cmd> ./build-key <name-of-user-or-group> </cmd>
为每一个用户重复使用相同的指令。

为密钥交换协议生成参数：
<cmd> ./build-dh </cmd>

提示：在这个指南里，服务器的名字直接使用 "server"，所以生成的钥匙分别被命名成 server.crt 和 server.key。同时我们的用户／用户组也被命名为 "client"。我们可以用任何的名字来命名，只要记住保存这些钥匙的文件路径。



# 服务器配置

接下来我们复制 "/etc/openvpn/sample-conf" 文件到 "/etc/openvpn/server.conf" 然后打开文件编辑配置。

提示：在行首的 ";" 表示该项已经被注释（没有效力）. 把 ";" 移除可以取消注释来生成效力。

编辑 server.conf 文件的通用提示：
我们可以设置一个没被别的服务占用的随机端口。端口 25 已经被 SMTP 占用，443 被 HTTPS 占用，22 被 SSH 占用。
我们通常需要一个被路由的 VPN，这样我们可以对 "dev tun" 项取消注释。参考 " 下载和安装 " 章节。使用 UDP 协议更好，因为比 TCP 更快。然后我们为 ca，cert 和 key 文件提供路径：
ca ca.crt
cert server.crt
key server.key

对密钥交换协议的设定我们需要提供协议的 pem，我们以前已经生成过了。
为子网络提供隧道的例子：
"Server 10.4.0.0 255.255.255.0"
参考 " 下载和安装 " 中的提示 2。