



Crie sua VPN com Banana Pi

Houve uma campanha publicitária através da internet sobre a vigilância do governo e espionagem. Uma dos melhores formas para proteger a sua privacidade digital é usando uma VPN para acessar a internet.

Embora os locais onde podemos colocar nossa Banana Pi são limitados, uma rede privada virtual também pode ajudar a passar por cima de firewalls em locais remotos ou conectar com segurança a sua rede doméstica de seu trabalho.

Open VPN

OpenVPN é um software opensource que permite a conexão segura a partir de um computador, smartphone, para um servidor. VPN, redes privadas virtuais, podem ser usados para hop sobre firewalls, acessar a internet sem restrições, ou ocultar o seu tráfego por trás de um servidor.

Com o Banana Pi M1, estaremos criando um servidor OpenVPN que irá fornecer o seu com uma VPN pessoal livre.

O que vamos usar?

Uma Banana Pi M1 com cartão SD e Bananian instalado
Ligação à Internet (com cabo)
Port Forwarding no router
Um computador / Smartphone

Todo o processo pode ser realizado com JuiceSSH no Android ao invés de usar um computador.

Fonte do Vídeo: <https://www.youtube.com/watch?v=6FoSoC4AA5o>

Outline

Fazer um servidor OpenVPN é um dos projetos mais difíceis aqui. Trata-se de muita configuração. Não é simplesmente copiar e colar o projeto! Fiz um esboço que mostra as etapas envolvidas.

- 1 - Preparando-se (atualização, alteração de senhas padrão e instalar OpenVPN)
- 2 - Gerando chaves (definir o tamanho da chave e gerar chaves criptografadas)
- 3 - Configuração do lado do servidor para a conexão
- 4 - Criando perfil **.ovpn** para uso do lado do cliente
- 5 - Port Forwarding no router
- 6 - Teste de Conexão

Começando o Projeto

Primeiro, você vai ter que atualizar seu sistema operacional para ter certeza que está atualizado e não vulnerável a bugs conhecidos.

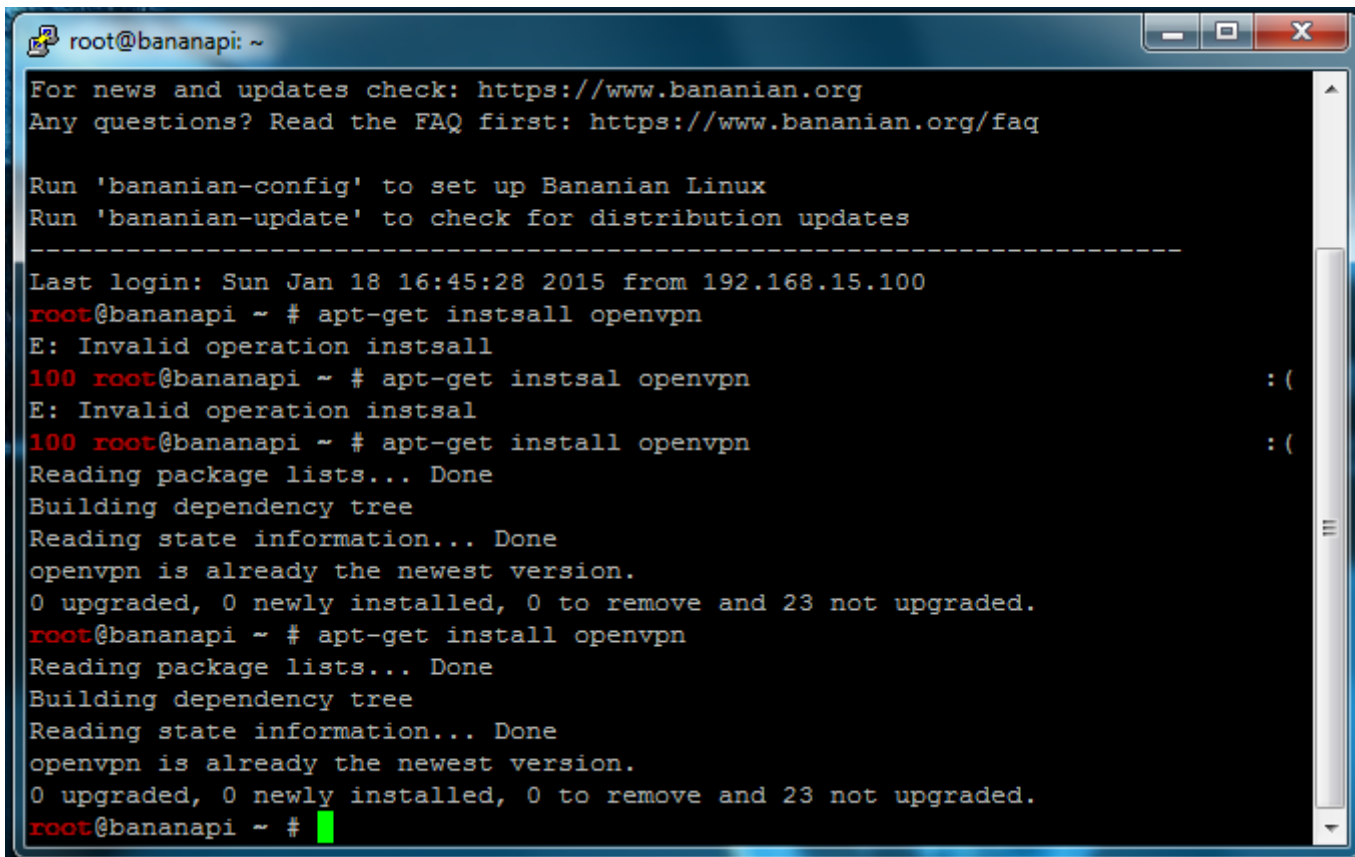
```
apt-get update  
apt-get upgrade
```

É altamente recomendável alterar a senha padrão, muitos roteadores foram hackeados porque as pessoas não se preocupam em mudar a senha padrão.

```
passwd
```

Em seguida, vamos instalar o OpenVPN

```
apt-get install openvpn
```



```
root@bananapi: ~  
For news and updates check: https://www.bananian.org  
Any questions? Read the FAQ first: https://www.bananian.org/faq  
  
Run 'bananian-config' to set up Bananian Linux  
Run 'bananian-update' to check for distribution updates  
-----  
Last login: Sun Jan 18 16:45:28 2015 from 192.168.15.100  
root@bananapi ~ # apt-get instsall openvpn  
E: Invalid operation instsall  
100 root@bananapi ~ # apt-get instsal openvpn  
E: Invalid operation instsal  
100 root@bananapi ~ # apt-get install openvpn  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
openvpn is already the newest version.  
0 upgraded, 0 newly installed, 0 to remove and 23 not upgraded.  
root@bananapi ~ # apt-get install openvpn  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
openvpn is already the newest version.  
0 upgraded, 0 newly installed, 0 to remove and 23 not upgraded.  
root@bananapi ~ # █
```

Gerando Chaves

Uma vez o OpenVPN instalado, temos que começar a gerar chaves e modificar as configurações. Em primeiro lugar, vamos fazer um novo diretório para as chaves.

```
mkdir /etc/openvpn/easy-rsa/  
cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0/* /etc/openvpn/easy-rsa/
```

Então, nós apenas copiamos alguma configuração exemplo para o diretório que será modificado e geramos as chaves.



Podemos continuar editando a configuração do servidor

Eu decidi mudar o tamanho da chave de criptografia de 1024 bits para 2048 bits para segurança extra.

```
sed -i 's/KEY_SIZE=1024/KEY_SIZE=2048/' /etc/openvpn/easy-rsa/vars
```

Agora, mude o diretório **easy-rsa** para aquele que acabamos de criar

```
nano /etc/openvpn/easy-rsa/vars
```

Altere o diretório theeasy-rsa do `pwd` para
/etc/openvpn/easy-rsa

```
root@bananapi: nano /etc/openvpn/easy-rsa/vars
GNU nano 2.2.6      File: /etc/openvpn/easy-rsa/vars

# easy-rsa parameter settings

# NOTE: If you installed from an RPM,
# don't edit this file in place in
# /usr/share/openvpn/easy-rsa --
# instead, you should copy the whole
# easy-rsa directory to another location
# (such as /etc/openvpn) so that your
# edits will not be wiped out by a future
# OpenVPN package upgrade.

# This variable should point to
# the top level of the easy-rsa
# tree.
export EASY_RSA="/etc/openvpn/easy-rsa"

#

# This variable should point to
# the requested executables

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Agora podemos começar a gerar as chaves. Precisamos apagar todas as chaves de exemplo e gerar chaves para o novo servidor.

Eu tive problemas no vídeo, mas agora este comando irá ajudá-lo muito! Dá-lhe-á permissão para executar programas na pasta.

```
chmod -R 777 /etc/openvpn/
cd /etc/openvpn/easy-rsa
source ./vars
./clean-all
```



Construindo as Chaves

```
./build-ca
```

Uma série de prompts irão aparecer, eu recomendo que você deixe tudo padrão e apenas clique em 'enter'. Se você mudar os padrões, tenha muito cuidado, pois as coisas podem não darem certo.

Construa uma chave do servidor. **'bananapi'** é o nome do meu servidor, você pode mudá-lo se quiser.

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

Gere outra chave:

```
openvpn --genkey --secret ta.key  
cd  
cp ca.key /etc/openvpn/easy-rsa/keys/ta.key
```

Crie uma chave privilegiada aleatória, ele vai levar um longo tempo. 10 a 20 minutos.

```
./build-dh
```

Construa uma chave para **user1**

```
./build-key-pass user1
```

Você pode mudar "user1" para qualquer nome que quiser, mas lembre-se que estamos criando perfis de usuário.

Quando solicitado, digite uma senha PEM de sua escolha, mas deixe tudo em branco!

Server Config

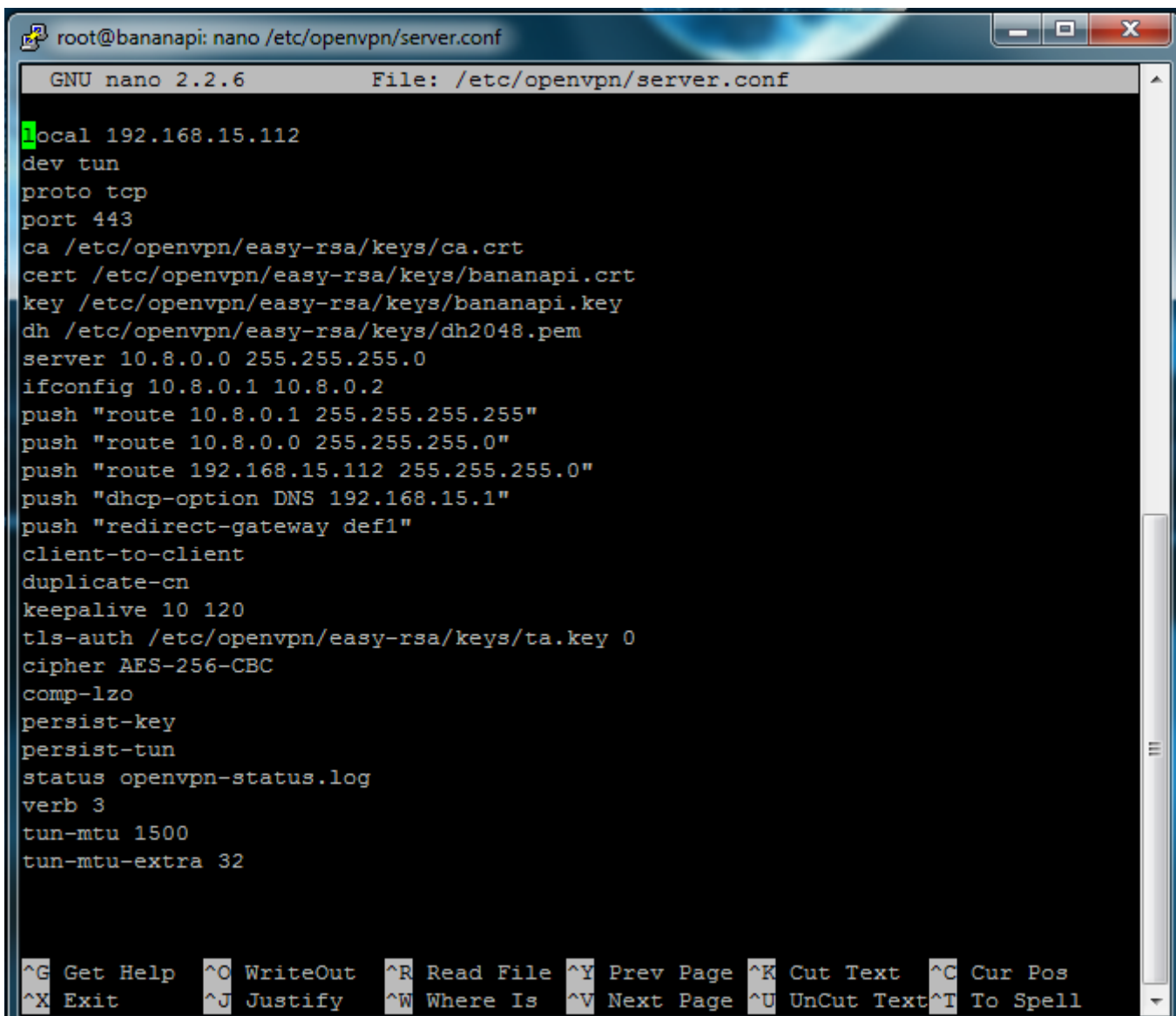
Agora, podemos proceder da seguinte forma: configure o servidor, uma vez que estaremos introduzindo linhas a partir do zero, pode ser uma boa idéia para você copiar e colar linhas abaixo:

```
nano /etc/openvpn/server.conf
```

Nano vai abrir um documento em branco, então cole as linhas abaixo nesse documento:

```
local 192.168.x.xxx  
#(fill in with local IP of your Banana Pi)  
dev tun  
proto tcp  
port 443  
ca /etc/openvpn/easy-rsa/keys/ca.crt  
cert /etc/openvpn/easy-rsa/keys/Server.crt  
key /etc/openvpn/easy-rsa/keys/Server.key  
dh /etc/openvpn/easy-rsa/keys/dh2048.pem  
server 10.8.0.0 255.255.255.0  
ifconfig 10.8.0.1 10.8.0.2  
push "route 10.8.0.1 255.255.255.255"  
push "route 10.8.0.0 255.255.255.0"  
push "route 192.168.x.xxx 255.255.255.0"  
#(fill in with Banana Pi IP)
```

```
push "dhcp-option DNS 192.168.x.x
#(fill in with your router IP)
push "redirect-gateway def1"
client-to-client
duplicate-cn
keepalive 10 120
tls-auth /etc/openvpn/easy-rsa/keys/ta.key 0
cipher AES-256-CBC
comp-lzo
persist-key
persist-tun
status openvpn-status.log
verb 3
```



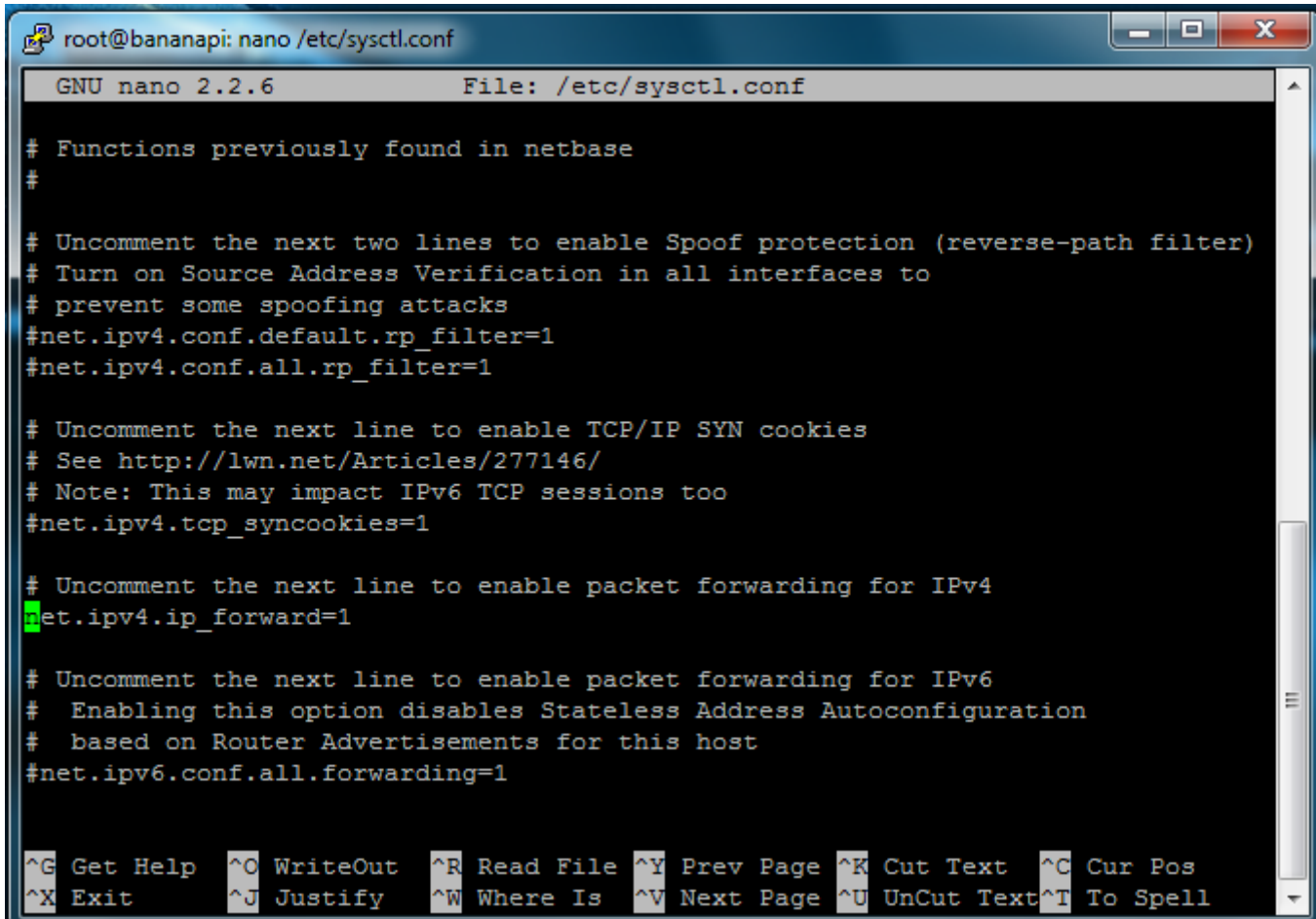
```
root@bananapi: nano /etc/openvpn/server.conf
GNU nano 2.2.6 File: /etc/openvpn/server.conf
local 192.168.15.112
dev tun
proto tcp
port 443
ca /etc/openvpn/easy-rsa/keys/ca.crt
cert /etc/openvpn/easy-rsa/keys/bananapi.crt
key /etc/openvpn/easy-rsa/keys/bananapi.key
dh /etc/openvpn/easy-rsa/keys/dh2048.pem
server 10.8.0.0 255.255.255.0
ifconfig 10.8.0.1 10.8.0.2
push "route 10.8.0.1 255.255.255.255"
push "route 10.8.0.0 255.255.255.0"
push "route 192.168.15.112 255.255.255.0"
push "dhcp-option DNS 192.168.15.1"
push "redirect-gateway def1"
client-to-client
duplicate-cn
keepalive 10 120
tls-auth /etc/openvpn/easy-rsa/keys/ta.key 0
cipher AES-256-CBC
comp-lzo
persist-key
persist-tun
status openvpn-status.log
verb 3
tun-mtu 1500
tun-mtu-extra 32
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Agora temos que habilitar o encaminhamento ipv4 no Banana Pi Server e modificar as configurações de firewall

nano /etc/sysctl.conf

Encontre a linha 'descomentar para permitir o encaminhamento de pacotes IPv4 "e excluir teh' #
'no início do script nessa linha.

nano /etc/firewall-openvpn-rules.sh



```
root@bananapi: nano /etc/sysctl.conf
GNU nano 2.2.6 File: /etc/sysctl.conf
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Estamos criando um novo arquivo que conterà as configurações. Preencha-o com as configurações abaixo:

```
#!/bin/bash
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j SNAT --to-source 192.168.XX.X
```

Nota: a linha vermelha é uma linha de comando.

```
root@bananapi: nano /etc/firewall-openvpn-rules.sh
GNU nano 2.2.6 File: /etc/firewall-openvpn-rules.sh
#!/bin/sh
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j SNAT --to-source 192.168.1.1
```

[Read 2 lines]

^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text	^C Cur Pos
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text	^T To Spell

Agora precisamos definir o arquivo para executar na inicialização

```
chmod 700 /etc/firewall-openvpn-rules.sh
chown root /etc/firewall-Openvpn-rules.sh
```

Fizemos o arquivo executável 'firewall-openvpn-rules.sh'

```
nano /etc/network/interfaces
```

Depois de "iface eth0 inet dhcp", travessão e adicione abaixo da linha como a imagem abaixo:
pre-up /etc/firewall-openvpn-rules.sh

```
root@bananapi: nano /etc/network/interfaces
GNU nano 2.2.6 File: /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto eth0

# dhcp configuration
iface eth0 inet dhcp
pre-up /etc/firewall-openvpn-rules.sh
# static ip configuration
#iface eth0 inet static
# address 192.168.6.241
# netmask 255.255.255.0
# gateway 192.168.6.1

[ Read 14 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Fizemos o arquivo `firewall-openvpn-rules.sh` executar automaticamente o caminho mais fácil. Depois, adicionamos a linha:

```
sh /etc/firewall-openvpn-rules.sh' after 'exit 0' at the end of the document /etc/rc.local
```

Agora é hora de criar um arquivo de configuração para o cliente abrir no Android. Crie um novo arquivo

```
nano /etc/openvpn/easy-rsa/keys/user1.ovpn
```

Vai abrir um documento em branco, então precisamos colocar o código abaixo:

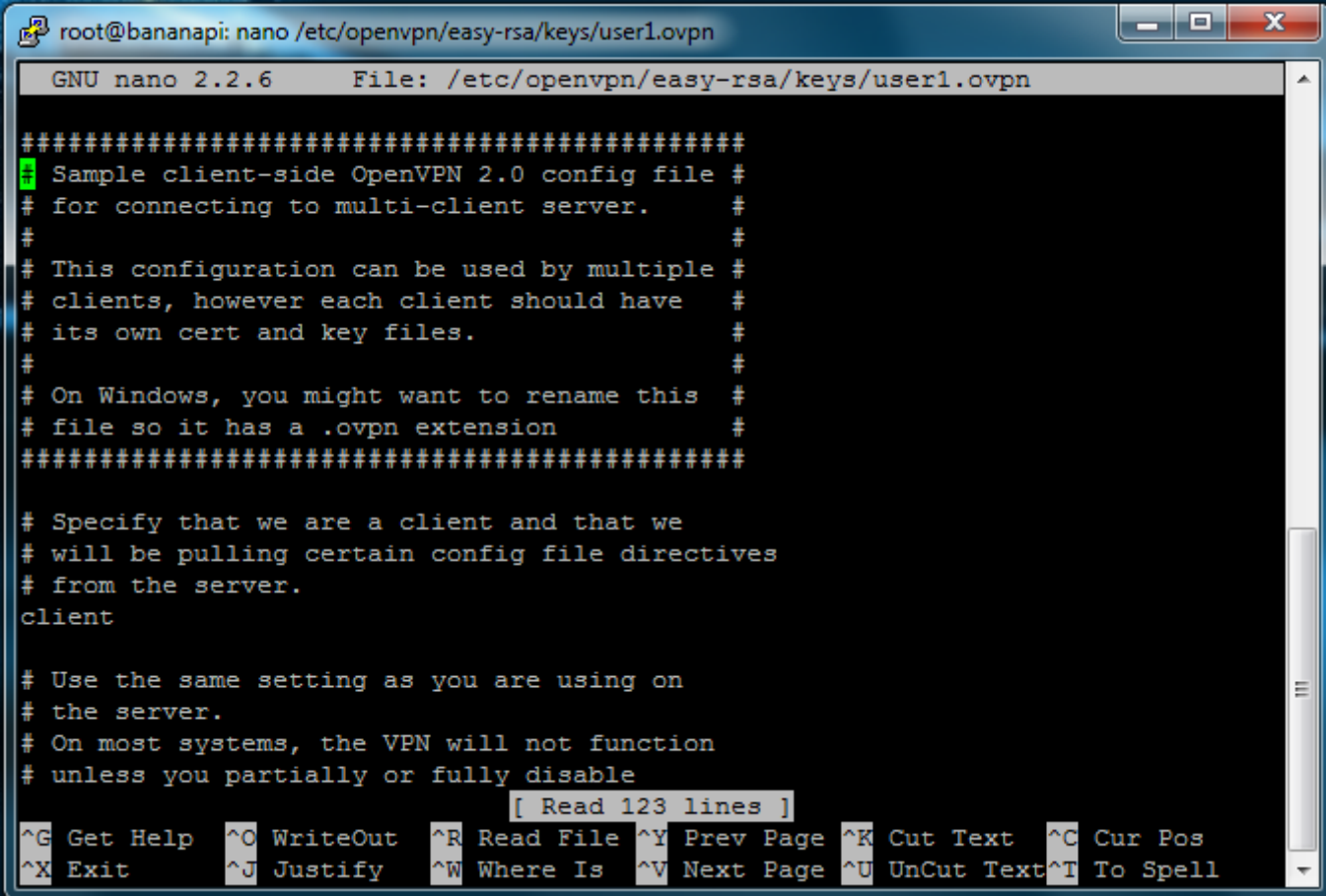
```
client
dev tun
proto tcp
remote (public IP of your house) 443
#(443 is the port)
resolv-retry infinite
persist-key
persist-tun
mute-replay-warnings
ca ca.crt
cert user1.crt
key user1.key
ns-cert-type server
```



```
cipher AES-256-CBC  
comp lzo  
verb3  
mute 20
```

Ou para ver o exemplo de configuração original, copie o arquivo conf como **user1.ovpn** (vermelho é uma linha)

```
cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn/easy-rsa/keys/user1.ovpn
```



```
root@bananapi: nano /etc/openvpn/easy-rsa/keys/user1.ovpn  
GNU nano 2.2.6 File: /etc/openvpn/easy-rsa/keys/user1.ovpn  
#####  
[1] Sample client-side OpenVPN 2.0 config file #  
# for connecting to multi-client server. #  
# #  
# This configuration can be used by multiple #  
# clients, however each client should have #  
# its own cert and key files. #  
# #  
# On Windows, you might want to rename this #  
# file so it has a .ovpn extension #  
#####  
  
# Specify that we are a client and that we  
# will be pulling certain config file directives  
# from the server.  
client  
  
# Use the same setting as you are using on  
# the server.  
# On most systems, the VPN will not function  
# unless you partially or fully disable  
[ Read 123 lines ]  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Agora use FileZilla para se conectar e baixar arquivos a partir da Banana Pi. Download '**ca.cert**'

```
user1.crt'user1.key'and'user1.ovpn
```

Depois que o arquivo está no seu PC, faça o upload para o Google Drive para que você possa baixá-lo em seu Android.

Start Open VPN and debug

Para iniciar e reiniciar o servidor:



```
service openvpn restart
```

Se o servidor não for iniciado, faça o debug com:

```
grep ovpn /var/log/syslog
```

Port Forwarding

Você já deve estar familiarizado e sabe que o encaminhamento de porta é fácil! Basta ir para a página de controle do seu roteador, digitando o seu endereço IP do roteador. O endereço IP deve ter essa aparência: **192.168.1.1**

Testando Conexão

Mova os arquivos do seu Banana Pi para o seu desktop ou nuvem com FileZilla usando sftp sob o diretório

```
/etc/openvpn/easy-rsa/keys/
```

arquivos para copiar:

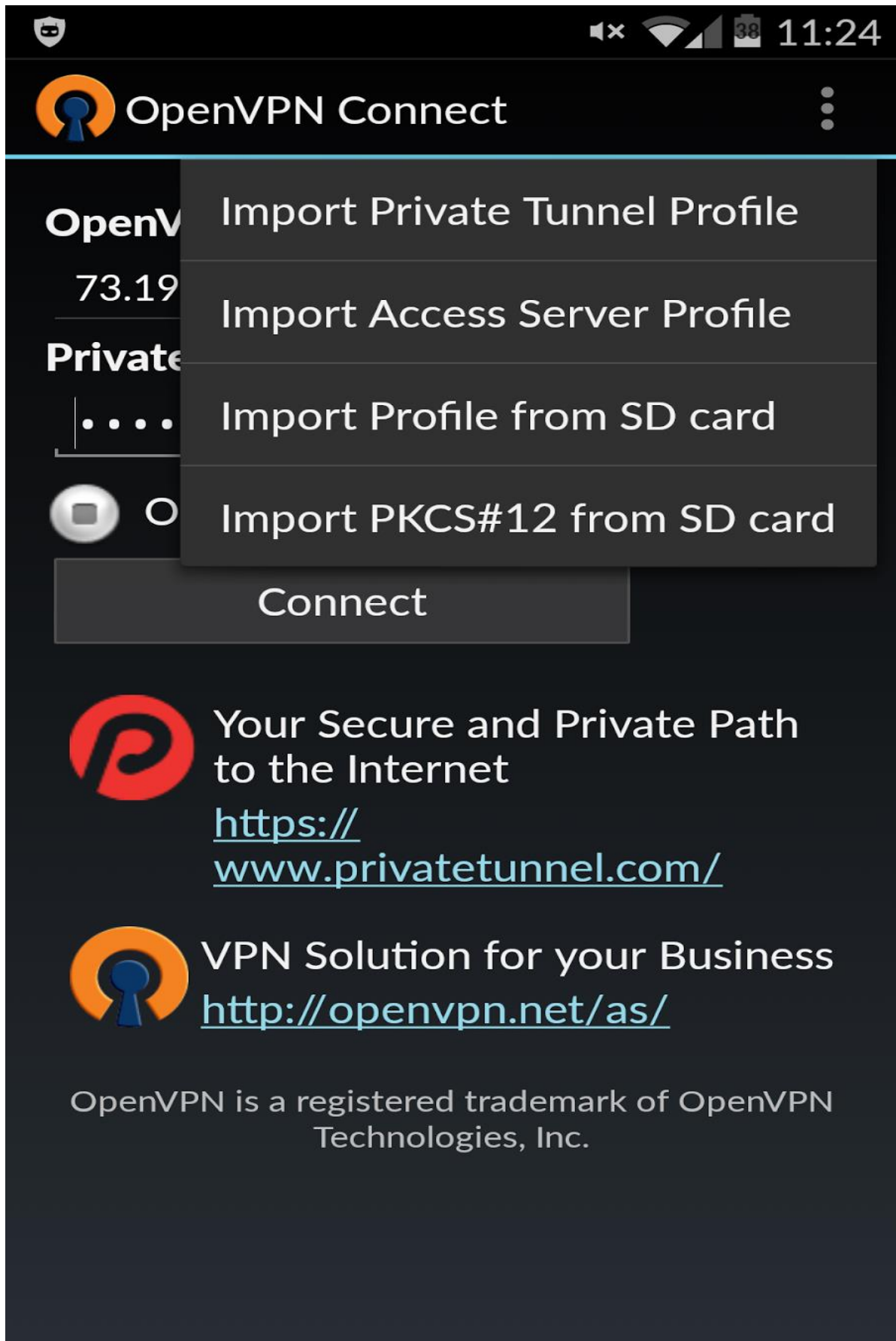
```
ca.crt  
user1.crt  
user1.key  
ta.key  
user1.ovpn
```

Firewall Jumping

Faça uma pesquisa no jumping firewall e veja se é um firewall básico, em seguida, defina a porta para 443 e protocolo TCP.

Mas há também packet sniffing firewalls que são mais difíceis de esconder, mas acho que um túnel SSL irá fazer o truque e vou fazer mais pesquisa e experiência com ele.

Agora podemos baixar OpenVPN conectado a partir do Play Store para testar a nossa VPN :)
Importe o arquivo **.ovpn** partir do cartão SD e está pronto!







Fonte: <http://projectbananapi.blogspot.com.br/2015/01/cheap-personal-vpn-with-banana-pi.html>