DigitalOcean

Subscribe

Como Configurar um Servidor OpenVPN no Ubuntu 16.04

6

Posted January 12, 2017 70.8k VPN UBUNTU UBUNTU 16.04

By: Justin Ellingwood

Introdução

Quer acessar a Internet com segurança a partir de seu smartphone ou laptop quando estiver conectado a uma rede não confiável, como o WiFi de um hotel ou café? Uma Virtual Private Network (VPN) permite que você percorra redes não confiáveis de forma privada e segura como se estivesse em uma rede privada. O tráfego emerge do servidor VPN e continua a sua viagem para o destino.

Quando combinada com conexões HTTPS, essa configuração permite que você proteja seus logins e transações sem fio. Você pode contornar restrições geográficas e censura, e proteger sua localização e qualquer tráfego HTTP não criptografado da rede não confiável.

OpenVPN é uma solução VPN Secure Socket Layer (SSL) repleta de recursos que aceita uma ampla gama de configurações. Nesse tutorial, vamos configurar um servidor OpenVPN em um Droplet e depois configurar o acesso a ele a partir do Windows, OS X, iOS e Android. Esse tutorial irá manter as etapas de instalação e configuração tão simples quanto possível para essas configurações.

Pré-requisitos

Para completar esse tutorial, você precisará de acesso a um servidor Ubuntu 16.04.

Você precisará configurar um usuário não-root com privilégios `sudo` antes de iniciar esse guia. Você pode seguir nosso guia de configuração inicial de servidor com Ubuntu 16.04 para configurar um usuário com as permissões apropriadas. O tutorial indicado também irá configurar um **firewall**, que assumiremos que já estará ativo durante esse guia.

Quando você estiver pronto para começar, acesse seu servidor como seu usuário `sudo` e continue abaixo.

Passo 1: Instalar o OpenVPN

Para começar, iremos instalar o OpenVPN em nosso servidor. O OpenVPN está disponível nos repositórios padrão do Ubuntu, portanto podemos usar o `apt` para a instalação. Estaremos instalando também o pacote `easy-rsa`, que nos ajudará a configurar um CA (certificate authority ou autoridade de certificação) interno para usar com nossa VPN.

Para atualizar o índice de pacotes de seu servidor e instalar os pacotes necessários, digite:

```
$ sudo apt-get update
$ sudo apt-get install openvpn easy-rsa
```

O software necessário está agora no servidor, pronto para ser configurado.

Passo 2: Configurar o Diretório CA

O OpenVPN é uma VPN TLS/SSL. Isso significa que ele utiliza certificados para criptografar o tráfego entre o servidor e os clientes. Para emitir certificados confiáveis, precisaremos configurar nossa própria autoridade de certificação (CA) simples.

Para começar, podemos copiar o diretório modelo `easy-rsa` em nosso diretório home com o comando `make-cadir`:

```
$ make-cadir ~/openvpn-ca
```

Mova-se para o diretório recém-criado para começar a configuração do CA:

```
$ cd ~/openvpn-ca
```

Passo 3: Configurar as Variáveis CA

Para configurar os valores que nossa CA irá utilizar, precisamos editar o arquivo `var` dentro do diretório. Abra esse arquivo agora em seu editor de textos:

```
$ nano vars
```

Dentro, você encontrará algumas variáveis que podem ser ajustadas para determinar como os seus certificados serão criados. Somente precisamos nos preocupar com algumas delas.

Na parte inferior do arquivo, localize as configurações que definem padrões de campo para novos certificados. Deve ser algo como isto:

~/openvpn-ca/vars

```
. . .

export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Fort-Funston"
export KEY_EMAIL="me@myhost.mydomain"
export KEY_OU="MyOrganizationalUnit"

. . .
```

Edite os valores em vermelho para o que você preferir, mas não os deixe em branco:

~/openvpn-ca/vars

```
. . .

export KEY_COUNTRY="US"
export KEY_PROVINCE="NY"
export KEY_CITY="New York City"
export KEY_ORG="DigitalOcean"
export KEY_EMAIL="admin@example.com"
export KEY_OU="Community"

. . .
```

Enquanto estamos aqui, também vamos editar o valor `KEY_NAME` logo abaixo dessa seção, que popula o campo de assunto. Para manter isso simples, vamos chamá-lo de `server` nesse guia:

~/openvpn-ca/vars

```
export KEY_NAME="server"
```

Quando tiver terminado, salve e feche o arquivo.

Passo 4: Construir a Autoridade de Certificação

Agora, podemos utilizar as variáveis que definimos e os utilitários `easy-rsa` para construir nossa autoridade de certificação.

Assegure-se de estar em seu diretório CA, e então carregue o arquivo `vars` que você acabou de editar:

```
$ cd ~/openvpn-ca
$ source vars
```

Você deve ver o seguinte se ele tiver sido carregado corretamente:

```
Output
NOTE: If you run ./clean-all, I will be doing a rm -rf on /home/sammy/openvpn-ca/keys
```

Certifique-se de que estamos operando em um ambiente limpo digitando:

```
$ ./clean-all
```

Agora, podemos construir nossa CA raiz digitando:

```
./build-ca
```

Isso iniciará o processo de criação da chave de autoridade de certificação raiz e do certificado. Como preenchemos o arquivo `vars` , todos os valores devem ser populados automaticamente. Basta pressionar **ENTER** através dos prompts para confirmar as seleções:

```
Output
Generating a 2048 bit RSA private key
.....................................................................................+
..............................+++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:
```

```
State or Province Name (full name) [NY]:
Locality Name (eg, city) [New York City]:
Organization Name (eg, company) [DigitalOcean]:
Organizational Unit Name (eg, section) [Community]:
Common Name (eg, your name or your server's hostname) [DigitalOcean CA]:
Name [server]:
Email Address [admin@email.com]:
```

Agora temos uma CA que pode ser utilizada para criar o restante dos arquivos que precisamos.

Passo 5: Criar o Certificado de Servidor, Chave e Arquivos de Criptografia

A seguir, vamos gerar nosso certificado de servidor e um par de chaves, bem como alguns arquivos adicionais utilizados durante o processo de criptografia.

Comece gerando o certificado do servidor OpenVPN e o par de chaves. Podemos fazer isso digitando:

> **Nota**: Se você escolher um nome diferente de `server` aqui, você terá que ajustar algumas das instruções abaixo. Por exemplo, quando copiar os arquivos gerados para o diretório `/etc/openvpn`, você terá que substituir os nomes corretos. Você também terá que modificar o arquivo `/etc/openvpn/server.conf` depois para apontar para os arquivos `.crt` e `.key` corretos.

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

Mais uma vez, os prompts terão valores padrão baseados nos argumentos que acabamos de passar em (`server`) e o conteúdo de nosso arquivo `vars` que carregamos.

Sinta-se livre para aceitar os valores padrão pressionando `ENTER`. *Não* insira uma senha de desafio para esta configuração. No final, você terá que digitar **y** para duas perguntas para assinar e confirmar o certificado:

```
Output
. . .

Certificate is to be certified until May  1 17:51:16 2026 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
```

A seguir, vamos gerar alguns outros itens. Podemos gerar chaves fortes Diffie-Hellman para usar durante a troca de chaves digitando:

```
$ ./build-dh
```

Isso pode levar alguns minutos para ser concluído.

Posteriormente, podemos gerar uma assinatura HMAC para fortalecer os recursos de verificação de integridade TLS do servidor:

```
$ openvpn --genkey --secret keys/ta.key
```

Passo 6: Gerar um Certificado Cliente e um Par de Chaves

A seguir, podemos gerar um certificado cliente e um par de chaves. Embora isso possa ser feito na máquina cliente e depois assinado pelo servidor/CA para propósitos de segurança, para esse guia vamos gerar a chave assinada no servidor por questões de simplicidade.

Vamos gerar um único certificado/chave para esse guia, mas se você tiver mais de um cliente, você pode repetir esse processo quantas vezes desejar. Passe um valor exclusivo para o script para cada cliente.

Como você pode voltar a essa etapa mais tarde, vamos recarregar o arquivo `vars`. Vamos utilizar `client1` como o valor para nosso primeiro par de certificado/chave para esse guia.

Para produzir credenciais sem uma senha, para ajudar em conexões automáticas, utilize o comando `build-key` dessa forma:

```
$ cd ~/openvpn-ca
$ source vars
$ ./build-key client1
```

Se, ao contrário, você desejar criar um conjunto de credencias protegidas por senha, utilize o comando `build-key-pass`:

```
$ cd ~/openvpn-ca
$ source vars
$ ./build-key-pass client1
```

Novamente, os padrões devem ser populados, assim basta pressionar **ENTER** para continuar. Deixe a senha de desafio em branco e certifique-se de inserir **y** para os prompts que pedem para assinar e confirmar o certificado.

Passo 7: Configurar o Serviço OpenVPN

A seguir, podemos começar a configuração do serviço OpenVPN utilizando as credenciais e arquivos que geramos.

Copiar os Arquivos para o Diretório OpenVPN

Para começar, precisamos copiar os arquivos que necessitamos para o diretório de configuração `/etc/openvpn` .

Podemos começar com todos os arquivos que acabamos de gerar. Eles foram colocados dentro do diretório `~/openvpn-ca/keys` quando foram criados. Precisamos mover o certificado e chave de nossa CA, o certificado e chave de nosso servidor, a assinatura HMAC, e o arquivo Diffie-Hellman.

```
$ cd ~/openvpn-ca/keys
$ sudo cp ca.crt ca.key server.crt server.key ta.key dh2048.pem /etc/openvpn
```

A seguir, precisamos copiar e descompactar um arquivo de configuração de exemplo do OpenVPN dentro do diretório de configuração para que possamos utilizá-lo como base para nossa configuração.

```
$ gunzip -c /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz | sudo tee /
```

Ajustar a Configuração do OpenVPN

Agora que nossos arquivos estão no lugar, podemos modificar o arquivo de configuração do servidor:

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

Configuração Básica

Primeiro, localize a seção HMAC olhando para a diretiva `tls-auth`. Remova o ";" para descomentar a linha `tls-auth`. Abaixo disso, adicione o parâmetro `key-direction` definido para "0":

/etc/openvpn/server.conf

```
tls-auth ta.key 0 # This file is secret
key-direction 0
```

Depois, localize a a seção sobre cifras criptográficas olhando para as linhas comentadas `cipher`. A cifra `AES-128-CBC` oferece um bom nível de criptografia e é bem suportada. Remova o ";" para descomentar a linha `cipher AES-128-CBC`:

/etc/openvpn/server.conf

```
cipher AES-128-CBC
```

Abaixo disso, adicione uma linha `auth` para selecionar algoritmo de resumo de mensagem HMAC. Para isso, `SHA256` é uma boa escolha:

/etc/openvpn/server.conf

```
auth SHA256
```

Finalmente, localize as configurações de `user` e `group` e remova o ";" do início para descomentar essas linhas:

/etc/openvpn/server.conf

```
user nobody
group nogroup
```

(Opcional) Forçar Alterações de DNS para Redirecionar Todo o Tráfego Através da VPN

As configurações acima irão criar a conexão VPN entre duas máquinas, mas não vai forçar quaisquer conexões para usarem o túnel. Se você deseja usar a VPN para rotear todo o seu tráfego, você provavelmente vai querer forçar as configurações de DNS para os computadores clientes.

Você pode fazer isso, descomentando algumas diretivas que vão configurar máquinas cliente para redirecionar todo o tráfego web através da VPN. Localize a seção `redirect-gateway` e remova o ponto e vírgula ";" do início da linha `redirect-gateway` para descomentá-la:

/etc/openvpn/server.conf

```
push "redirect-gateway def1 bypass-dhcp"
```

Logo abaixo disso, localize a seção `dhcp-option`. Novamente, remova o ";" na frente de ambas as linhas para descomentá-las:

/etc/openvpn/server.conf

```
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"
```

Isso deve ajudar os clientes a reconfigurar suas configurações de DNS para utilizar o túnel VPN como gateway padrão.

(Opcional) Ajustar a Porta e o Protocolo

Por padrão, o servidor OpenVPN usa a porta 1194 e o protocolo UDP para aceitar conexões de clientes. Se você precisar usar uma porta diferente devido a ambientes de redes restritivos onde seus clientes podem estar, você pode alterar a opção `port`. Se você não está hospedando conteúdo web em seu servidor OpenVPN, a porta 443 é uma escolha popular, uma vez que ela é permitida através das regras de firewall.

/etc/openvpn/server.conf

```
# Optional!
port 443
```

Muitas vezes o protocolo estará restrito a essa porta também. Se assim for, altere `proto` de UDP para TCP:

/etc/openvpn/server.conf

```
# Optional!
proto tcp
```

Se você não tem necessidade de utilizar uma porta diferente, é melhor deixar essas duas configurações como padrão.

(Opcional) Apontar para Credenciais Não-Padrão

Se você selecionou um nome diferente durante o comando `./build-key-server` mais cedo, modifique as linhas `cert` e `key` que você vê apontar para os arquivos `.crt` e `.key` apropriados. Se você utilizou o padrão `server`, isso já deve estar definido corretamente:

/etc/openvpn/server.conf

```
cert server.crt
key server.key
```

Quando tiver terminado, salve e feche o arquivo.

Passo 8: Ajustar a Configuração de Rede do Servidor

A seguir, precisamos ajustar alguns aspectos da rede do servidor para que o OpenVPN possa rotear o tráfego corretamente.

Permitir o Encaminhamento IP

Primeiro, percisamos permitir ao servidor encaminhar o tráfego. Isso é essencial para a funcionalidade que queremos que nosso servidor de VPN forneça.

Podemos ajustar essa configuração modificando o arquivo `/etc/sysctl.conf` :

```
$ sudo nano /etc/sysctl.conf
```

Dentro dele, olhe para a linha que define `net.ipv4.ip_forward` . Remova o caractere "#" do início da linha para descomentar essa configuração:

/etc/sysctl.conf

```
net.ipv4.ip_forward=1
```

Salve e feche o arquivo quando tiver terminado.

Para ler o arquivo e ajustar os valores para a sessão atual, digite:

```
$ sudo sysctl -p
```

Ajustar as Regras UFW para Mascarar Conexões de Clientes

Se você seguiu o guia de configuração inicial de servidor com Ubuntu 16.04 nos pré-requisitos, você deve ter o firewall UFW ativo. Independentemente de você usar o firewall para bloquear tráfego indesejado (que você quase sempre deve fazer), precisamos do firewall nesse guia para manipular parte do tráfego que entra no servidor. Precisamos modificar o arquivo de regras para configurar o mascaramento, um conceito `iptables` que fornece NAT dinâmico sob demanda para rotear corretamente as conexões de clientes.

Antes de abrirmos o arquivo de configuração do firewall para adicionar o mascaramento, precisamos encontrar a interface pública de rede de nossa máquina. Para fazer isso, digite:

```
$ ip route | grep default
```

Sua interface pública deve seguir a palavra "dev". Por exemplo, esse resultado mostra a interface chamada `wlp11s0`, que está destacada abaixo:

```
Output
default via 203.0.113.1 dev wlp11s0  proto static  metric 600
```

Quando você tiver a interface associada com sua rota padrão, abra o arquivo `/etc/ufw/before.rules` para adicionar a configuração relevante:

```
$ sudo nano /etc/ufw/before.rules
```

Esse arquivo trata da configuração que deve ser acionada antes que as regras UFW convencionais sejam carregadas. Na parte superior do arquivo, adicione as linhas destacadas abaixo: Isso irá definir a política padrão para o canal `POSTROUTING` na tabela `nat` e mascarar qualquer tráfego vindo da VPN:

> **Nota**: Lembre-se de substituir `eth0` na linha `-A POSTROUTING` abaixo com a interface que você encontrou no comando acima.

/etc/ufw/before.rules

```
#
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#

# START OPENVPN RULES
# NAT table rules
*nat
```

```
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to eth0
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
COMMIT
# END OPENVPN RULES

# Don't delete these required lines, otherwise there will be errors
*filter
. . .
```

Salve e feche o arquivo quando terminar.

Precisamos dizer ao UFW para permitir pacotes encaminhados também. Para fazer isso, vamo abrir o arquivo `/etc/default/ufw`:

```
$ sudo nano /etc/default/ufw
```

Dentro dele, localize a diretiva `DEFAULT_FORWARD_POLICY`. Vamos alterar o valor de `DROP` para `ACCEPT`:

/etc/default/ufw

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Salve e feche o arquivo ao terminar.

Abrir a Porta do OpenVPN e Habilitar as Alterações

A seguir, vamos ajustar o firewall em si para permitir tráfego para o OpenVPN.

Se você não modificou a porta e o protocolo no arquivo `/etc/openvpn/server.conf`, você vai precisar abrir o tráfego UDP para a porta 1194. Se você modificou a porta e/ou o protocolo, substitua os valores que você escolheu aqui.

Também adicionaremos a porta SSH caso você tenha esquecido de adicioná-la ao seguir o tutorial de pré-requisito:

```
$ sudo ufw allow 1194/udp
$ sudo ufw allow OpenSSH
```

Agora, podemos desabilitar e re-habilitar o UFW para carregar as alterações de todos os arquivos que modificamos:

```
$ sudo ufw disable
$ sudo ufw enable
```

Agora, nosso servidor está configurado para tratar corretamente o tráfego OpenVPN.

Passo 9: Iniciar e Habilitar o Serviço OpenVPN

Estamos finalmente prontos para iniciar o serviço OpenVPN em nosso servidor. Podemos fazer isso usando o systemd.

Precisamos iniciar o servidor OpenVPN especificando o nome do nosso arquivo de configuração como uma variável de instância após o nome do arquivo de unidade systemd. Nosso arquivo de configuração para nosso servidor é chamado `/etc/openvpn/server.conf`, assim vamos adicionar `@server` ao final de nosso arquivo de unidade ao chamá-lo:

```
$ sudo systemctl start openvpn@server
```

Verifique novamente se o serviço foi iniciado com êxito, digitando:

```
$ sudo systemctl status openvpn@server
```

Se tudo correu bem, sua saída deve ser algo parecido com isso:

```
Output
● openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2016-05-03 15:30:05 EDT; 47s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn23ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
  Process: 5852 ExecStart=/usr/sbin/openvpn --daemon ovpn-%i --status /run/openvpn/%i.statu
 Main PID: 5856 (openvpn)
    Tasks: 1 (limit: 512)
   CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
           └─5856 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.statu

May 03 15:30:05 openvpn2 ovpn-server[5856]: /sbin/ip addr add dev tun0 local 10.8.0.1 peer
May 03 15:30:05 openvpn2 ovpn-server[5856]: /sbin/ip route add 10.8.0.0/24 via 10.8.0.2
May 03 15:30:05 openvpn2 ovpn-server[5856]: GID set to nogroup
May 03 15:30:05 openvpn2 ovpn-server[5856]: UID set to nobody
May 03 15:30:05 openvpn2 ovpn-server[5856]: UDPv4 link local (bound): [undef]
May 03 15:30:05 openvpn2 ovpn-server[5856]: UDPv4 link remote: [undef]
```

```
May 03 15:30:05 openvpn2 ovpn-server[5856]: MULTI: multi_init called, r=256 v=256
May 03 15:30:05 openvpn2 ovpn-server[5856]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0
May 03 15:30:05 openvpn2 ovpn-server[5856]: IFCONFIG POOL LIST
May 03 15:30:05 openvpn2 ovpn-server[5856]: Initialization Sequence Completed
```

Você também pode verfificar que a interface OpenVPN `tun0` está disponível digitando:

```
$ ip addr show tun0
```

Você deve ver uma interface configurada:

```
Output
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN grc
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
       valid_lft forever preferred_lft forever
```

Se tudo correu bem, habilite o serviço para que ele inicie automaticamente no boot:

```
$ sudo systemctl enable openvpn@server
```

Passo 10: Criar Infraestrutura de Configuração de Cliente

A seguir, precisamos configurar um sistema que nos permitirá criar arquivos de configuração de cliente facilmente.

Criando a Estrutura de Diretório de Configuração do Cliente

Crie uma estrutura de diretório dentro do seu diretório home para armazenar os arquivos:

```
$ mkdir -p ~/client-configs/files
```

Como nossos arquivos de configuração de cliente terão as chaves de cliente embutidas, devemos bloquear as permissões em nosso diretório interno:

```
$ chmod 700 ~/client-configs/files
```

Criando uma Configuração Básica

Em seguida, vamos copiar um exemplo de configuração de cliente em nosso diretório para usar como nossa configuração base:

```
$ cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-configs/base
```

Abra esse novo arquivo em seu editor de textos:

```
$ nano ~/client-configs/base.conf
```

Dentro dele, precisamos fazer alguns ajustes.

Primeiro, localize a diretiva `remote`. Isso aponta o cliente para o endereço do nosso servidor OpenVPN. Este deve ser o endereço IP público do nosso servidor OpenVPN. Se você mudou a porta na qual o servidor OpenVPN está escutando, mude `1194` para a porta que você escolheu:

~/client-configs/base.conf

```
. . .
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote endereço_IP_do_servidor 1194
. . .
```

Certifique-se de que o protocolo corresponde ao valor que você está utilizando na configuração do servidor:

~/client-configs/base.conf

```
proto udp
```

Em seguida, descomente as diretivas `user` e `group` removendo o ";":

~/client-configs/base.conf

```
# Downgrade privileges after initialization (non-Windows only)
user nobody
group nogroup
```

Encontre as diretivas que definem `ca`, `cert`, e `key`. Comente essas diretivas já que vamos adicionar os certificados e as chaves dentro do próprio arquivo:

~/client-configs/base.conf

```
# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
#ca ca.crt
#cert client.crt
#key client.key
```

Espelhe as configurações `cipher` e `auth` que definimos no arquivo `/etc/openvpn/server.conf`:

~/client-configs/base.conf

```
cipher AES-128-CBC
auth SHA256
```

Depois, adicione a diretiva `key-direction` em algum lugar no arquivo. Isso deve estar definido para "1" para funcionar com o servidor:

~/client-configs/base.conf

```
key-direction 1
```

Finalmente, adicione algumas linhas **comentadas**. Queremos incluir essas linhas em toda configuração, mas somente devem ser habilitadas para clientes Linux que vêm com um arquivo `/etc/openvpn/update-resolv-conf`. Esse script usa o utilitário `resolvconf` para atualizar informações DNS para clientes Linux.

~/client-configs/base.conf

```
# script-security 2
# up /etc/openvpn/update-resolv-conf
# down /etc/openvpn/update-resolv-conf
```

Se seu cliente está usando Linux e tem um arquivo `/etc/openvpn/update-resolv-conf`, você deve descomentar essas linhas do arquivo de configuração de cliente OpenVPN que foi gerado.

Salve o arquivo quando tiver terminado.

Criando um Script de Geração de Configuração

A seguir, vamos criar um script simples para compilar nossa configuração básica com o certificado relevante, chave, e arquivos de criptografia. Ele irá colocar os arquivos de configuração gerados no diretório `~/client-configs/files`.

Crie e abra um arquivo chamado `make_config.sh` dentro do diretório `~/client-configs`:

```
$ nano ~/client-configs/make_config.sh
```

Dentro, cole o seguinte script:

~/client-configs/make_config.sh

```
#!/bin/bash


# First argument: Client identifier

KEY_DIR=~/openvpn-ca/keys
OUTPUT_DIR=~/client-configs/files
BASE_CONFIG=~/client-configs/base.conf

cat ${BASE_CONFIG} \
    <(echo -e '<ca>') \
    ${KEY_DIR}/ca.crt \
    <(echo -e '</ca>\n<cert>') \
    ${KEY_DIR}/${1}.crt \
    <(echo -e '</cert>\n<key>') \
    ${KEY_DIR}/${1}.key \
    <(echo -e '</key>\n<tls-auth>') \
    ${KEY_DIR}/ta.key \
    <(echo -e '</tls-auth>') \
    > ${OUTPUT_DIR}/${1}.ovpn
```

Salve e feche o arquivo quando tiver terminado.

Marque o arquivo como executável digitando:

```
$ chmod 700 ~/client-configs/make_config.sh
```

Passo 11: Gerar Configurações de Cliente

Agora, podemos gerar facilmente arquivos de configuração de cliente.

Se você acompanhou o guia, você criou um certificado de cliente e uma chave chamados `client1.crt` e `client1.key` respectivamente executando o comando `./build-key client1` no passo 6. Podemos gerar uma configuração para essas credenciais movendo-as para dentro de nosso diretório `~/client-configs` e utilizando o script que fizemos:

```
$ cd ~/client-configs
$ ./make_config.sh client1
```

Se tudo correu bem, devemos ter um arquivo `client1.ovpn` em nosso diretório `~/client-configs/files`:

```
$ ls ~/client-configs/files
```

```
Output
client1.ovpn
```

Transferindo a Configuração para Dispositivos Cliente

Precisamos transferir o arquivo de configuração de cliente para o dispositivo relevante. Por exemplo, pode ser seu computador ou um dispositivo móvel.

Embora os aplicativos exatos usados para realizar essa transferência dependem de sua escolha e do sistema operacional do dispositivo, deseja-se que a aplicação utilize SFTP (Protocolo de Transferência de Arquivo SSH) ou SCP (Cópia Segura) na retaguarda. Isso transportará os arquivos de autenticação VPN do cliente por meio de uma conexão criptografada.

Aqui está um exemplo de comando SFTP utilizando nosso exemplo client1.ovpn. Esse comando pode ser executado a partir do seu computador local (OS X or Linux). Ele coloca o arquivo `.ovpn` em nosso diretório home:

```
$ sftp sammy@ip_servidor_openvpn:client-configs/files/client1.ovpn ~/
```

Aqui estão várias ferramentas e tutoriais para transferir arquivos com segurança do servidor para uma computador local:

- WinSCP
- How To Use SFTP to Securely Transfer Files with a Remote Server
- How To Use Filezilla to Transfer and Manage Files Securely on your VPS

Passo 12: Instalar a Configuração do Cliente

Agora, vamos discutir como instalar um perfil de cliente VPN no Windows, OS X, iOS, e Android. Nenhuma dessas instruções de cliente são dependentes uma da outra, portanto, sinta-se livre para pular para a que for aplicável a você.

A conexão OpenVPN será chamada do que quer que você nomeou o arquivo `.ovpn`. Em nosso exemplo, isso significa que a conexão será chamada `client1.ovpn` para o primeiro arquivo de cliente que geramos.

Windows

Instalando

A aplicação OpenVPN cliente para Windows pode ser encontrada em OpenVPN's Downloads page. Escolha a versão de instalador apropriada para sua versão de Windows.

Nota

```
OpenVPN needs administrative privileges to install.
```

Depois da instalação do OpenVPN, copie o arquivo `.ovpn` para:

```
C:\Program Files\OpenVPN\config
```

Ao iniciar o OpenVPN, ele verá automaticamente o perfil e o torna disponível.

O OpenVPN deve ser executado como um administrador cada vez que é utilizado, mesmo por contas administrativas. Para fazer isso sem ter que clicar com o botão direito do mouse e selecionar **Executar como Administrador** toda vez que usar a VPN, você pode predefinir isso, mas isso tem que ser feito a partir de uma conta administrativa. Isso também significa que usuários comuns precisarão inserir a senha do administrador para usar o OpenVPN. Por outro lado, usuários comuns não podem se conectar apropriadamente ao servidor a menos que a aplicação OpenVPN no cliente tenha direitos administrativos, assim os privilégios elevados são necessários.

Para configurar a aplicação OpenVPN para sempre executar como administrador, clique com o botão direito do mouse no seu ícone de atalho e vá em **Propriedades**. Na parte inferior da aba **Compatibilidade**, clique no botão para **Alterar configurações de todos os usuários**. Na nova janela, marque **Executar este programa como Administrador**.

Conectando

Cada vez que você iniciar a OpenVPN GUI, o Windows perguntará se você deseja permitir que o programa faça alterações em seu computador. Clique em **Sim**. Iniciar o aplicativo cliente OpenVPN somente coloca o applet na bandeja do sistema para que a VPN possa ser conectada e desconectada conforme necessário; ele na verdade não faz a conexão VPN.

Um vez que o OpenVPN é iniciado, inicie uma conexão indo para o applet da bandeja do sistema e clique com o botão direito do mouse no ícone do applet OpenVPN. Isso abre o menu de contexto. Selecione **client1** no topo no menu (que é o nosso perfil `client1.ovpn`) e escolha **Connect**.

Uma janela de status irá se abrir mostrando a saída de log enquanto a conexão é estabelecida, e uma mensagem será exibida quando o cliente estiver conectado.

Desconecte-se da VPN da mesma forma: Vá para o applet da bandeja do sistema, clique com o botão direito do mouse no ícone do applet OpenVPN, selecione o perfil de cliente e clique em **Disconnect**.

OS X

Instalando

Tunnelblick é um cliente OpenVPN de código aberto, gratuito para Mac OS X. Você pode baixar a última imagem de disco a partir de Tunnelblick Downloads page. Dê um clique duplo no arquivo `.dmg` baixado e siga as instruções para instalar.

No final do processo de instalação, o Tunnelblick irá perguntar se você tem quaisquer arquivos de configuração. Pode ser mais fácil responder **No** e deixar o Tunnelblick terminar. Abra uma janela do Finder e dê um clique duplo em `client1.ovpn`. O Tunnelblick vai instalar o perfil do cliente. Privilégios administrativos são necessários.

Conectando

Inicie o Tunnelblick dando um duplo clique no ícone do aplicativo na pasta **Applications**. Uma vez que o Tunnelblick foi iniciado, haverá um ícone do programa na barra de menu na parte superior direita da tela para controle de conexões. Clique no ícone, e depois no item de menu **Connect** para iniciar a conexão VPN. Selecione a conexão **client1**.

Linux

Instalando

Se você estiver usando Linux, existe uma variedade de ferramentas que você pode utilizar dependendo da sua distribuição. Seu ambiente desktop ou gerenciador de janelas também pode incluir utilitários de conexão.

A maneira mais universal de conexão, contudo, é apenas usar o software OpenVPN.

No Ubuntu ou Debian, você pode instalá-lo da mesma forma que você fez no servidor digitando:

```
client$ sudo apt-get update
client$ sudo apt-get install openvpn
```

No CentOS você pode habilitar os repositórios EPEL e então instalá-lo digitando:

```
client$ sudo yum install epel-release
client$ sudo yum install openvpn
```

Configurando

Verifique para ver se sua distribuição inclui um script `/etc/openvpn/update-resolv-conf`:

```
client$ ls /etc/openvpn
```

```
Output
update-resolve-conf
```

Depois, edite a configuração do arquivo de cliente OpenVPN que você transferiu:

```
client$ nano client1.ovpn
```

Descomente as três linhas que colocamos para ajustar as configurações DNS se você for capaz de encontrar um arquivo `update-resolv-conf`:

client1.ovpn

```
script-security 2
up /etc/openvpn/update-resolv-conf
down /etc/openvpn/update-resolv-conf
```

Se você está usando CentOS, mude o `group` de `nogroup` para `nobody` para corresponder aos grupos disponíveis na distribuição:

client1.ovpn

```
group nobody
```

Salve e feche o arquivo:

Agora, você pode se conectar à VPN simplesmente apontando o comando `openvpn` para o arquivo de configuração do cliente:

```
client$ sudo openvpn --config client1.ovpn
```

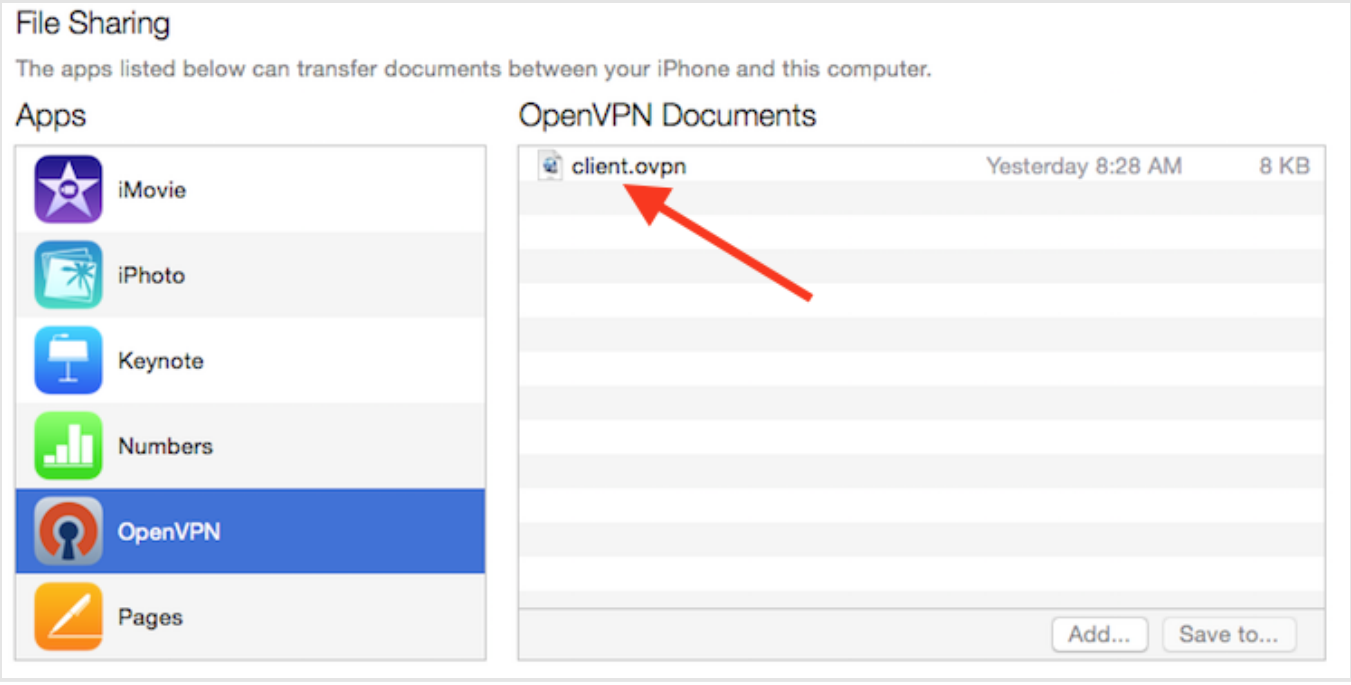
Isso deve conectar você ao seu servidor.

iOS

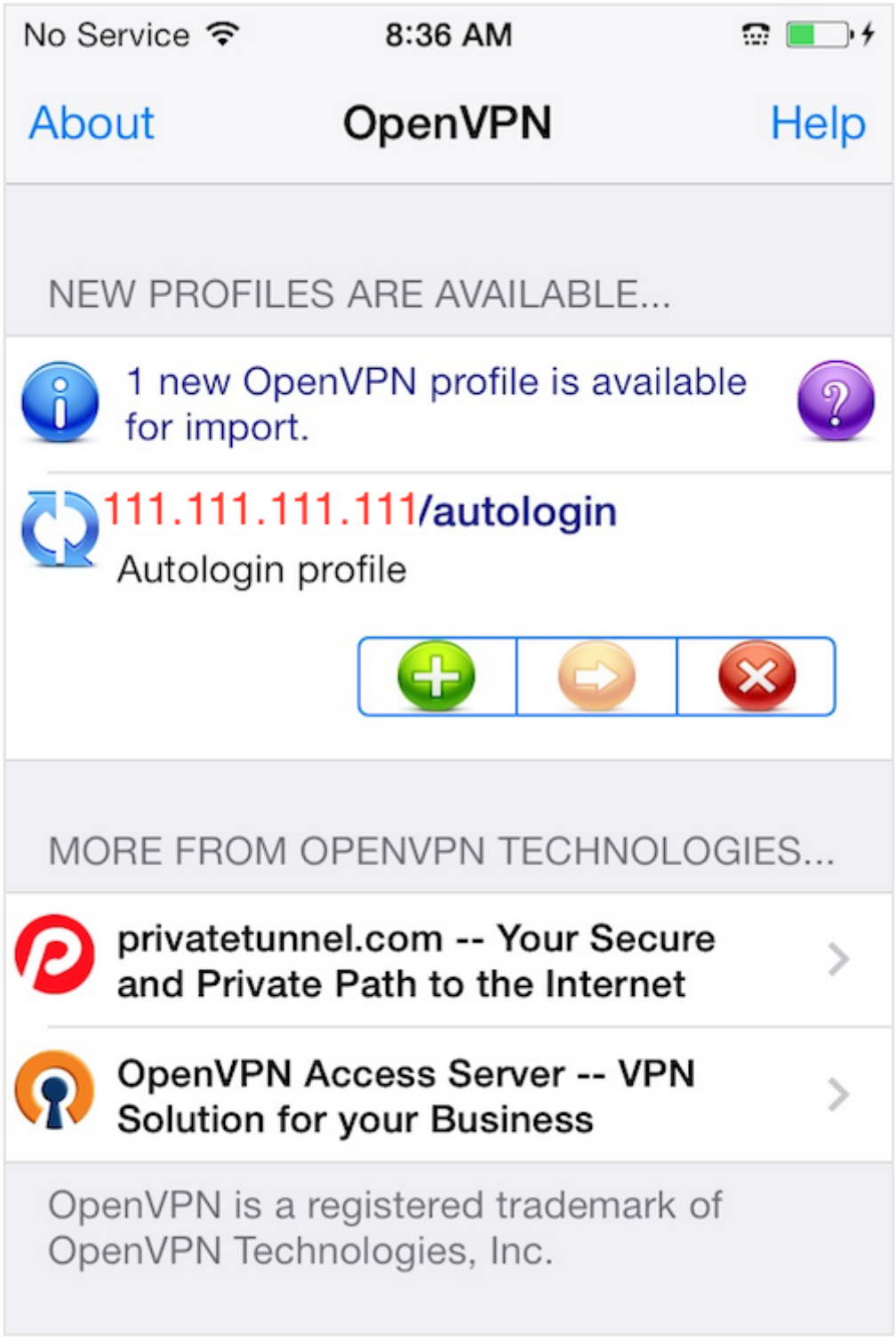
Instalando

A partir do iTunes App Store, pesquise por OpenVPN Connect e instale a aplicação oficial OpenVPN para iOS. Para transferir a configuração do cliente iOS para o dispositivo, conecte-o diretamente a um computador.

A conclusão da transferência com o iTunes será descrita aqui. Abra o iTunes no computador e clique em **iPhone** > **apps**. Role para baixo até a parte inferior na seção **File Sharing** e clique no app OpenVPN. A janela em branco à direita, **OpenVPN Documents**, é para compartilhamento de arquivos. Arraste o arquivo `.ovpn` para a janela Documents do OpenVPN.



Agora, inicie o app OpenVPN no iPhone. Haverá uma notificação de que um novo perfil está pronto para ser importado. Toque no sinal verde de adição para importá-lo.

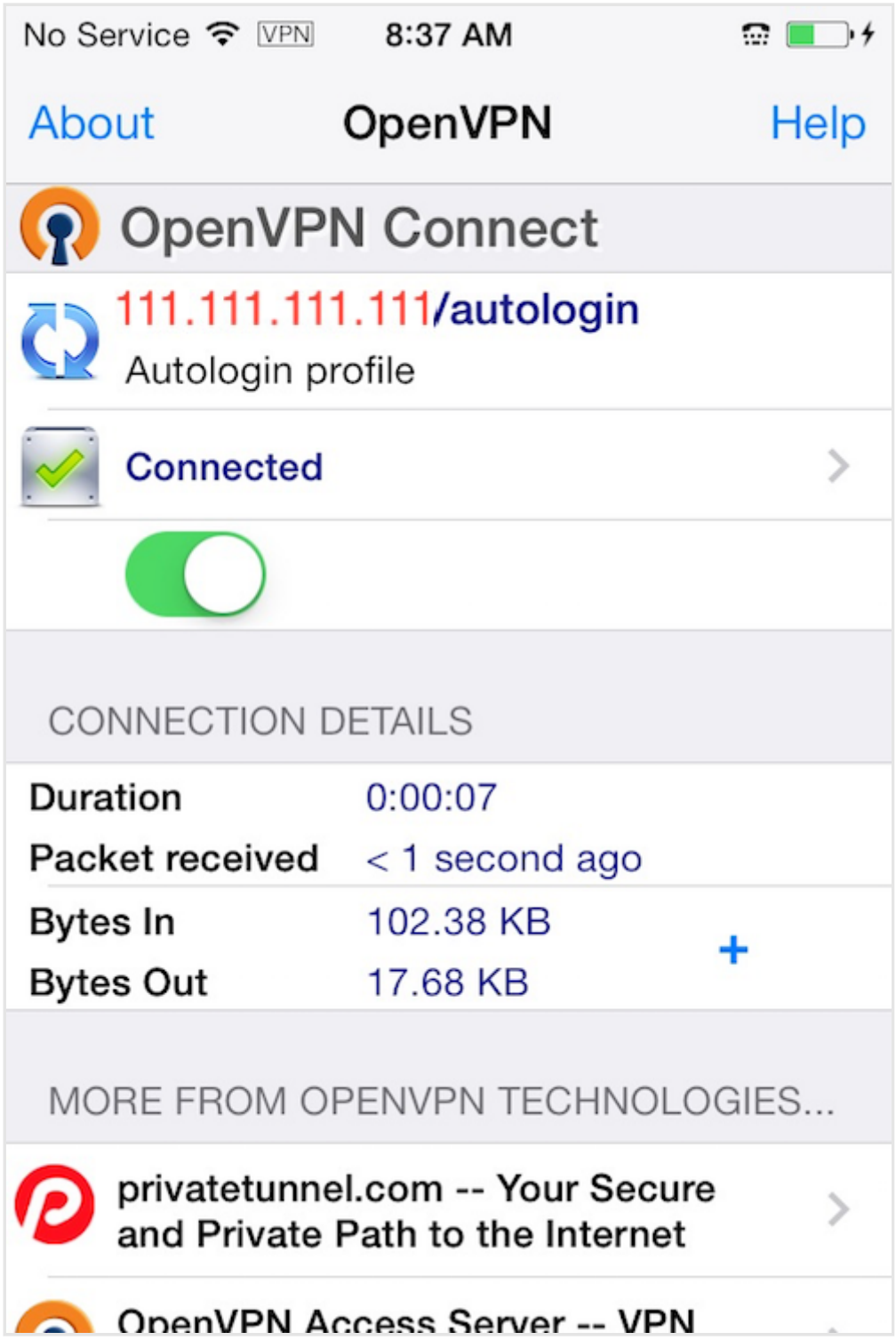


Conectando

O OpenVPN está agora pronto para ser utilizado com o novo perfil. Inicie a conexão deslizando o botão **Connect** para a posição **On**. Desconecte deslizando o mesmo botão para **Off**.

Nota

```
A chave VPN abaixo de **Settings** não pode ser utilizada para conectar na VPN. Se você ter
```



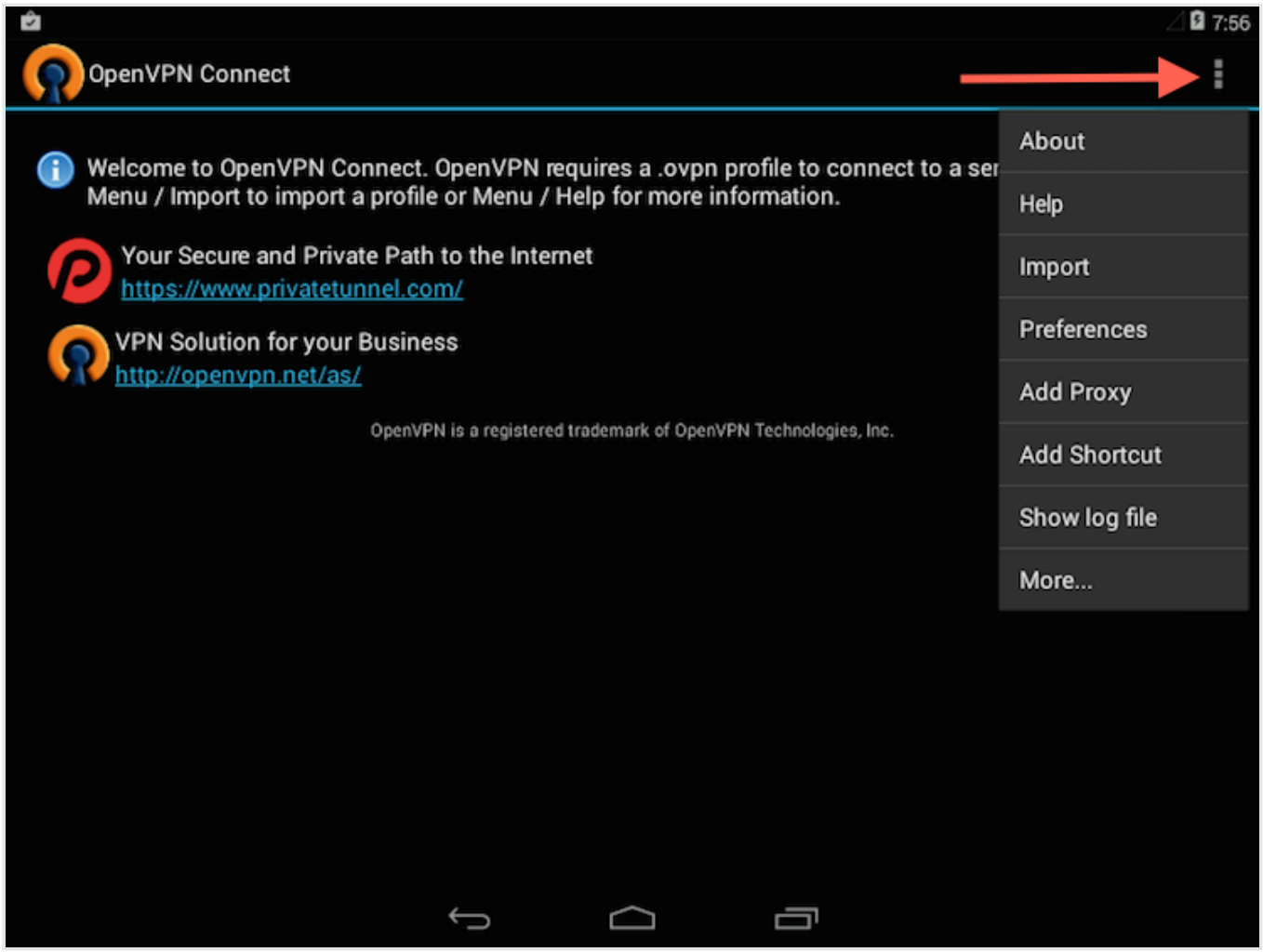
Android

Instalando

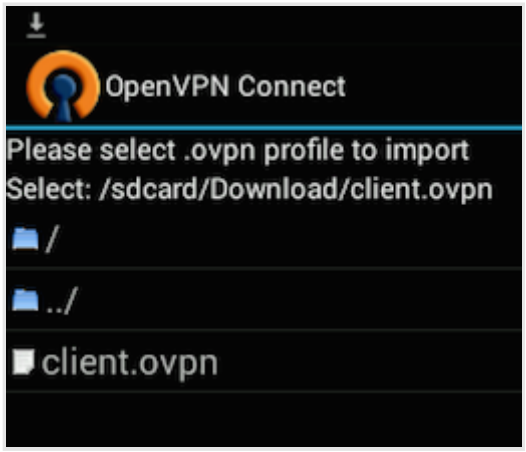
Abra o Google Play Store. Pesquise por Android OpenVPN Connect e instale a aplicação oficial OpenVPN para Android.

O perfil `.ovpn` pode ser transferido conectando-se o dispositivo Android ao seu computador pelo USB e copiando o arquivo. Alternativamente, se você tiver um leitor de cartões SD, você pode remover o cartão SD do dispositivo, copiar o perfil nele e então, inserir o cartão de volta no dispositivo Android.

Inicie o app OpenVPN e toque o menu para importar o perfil.

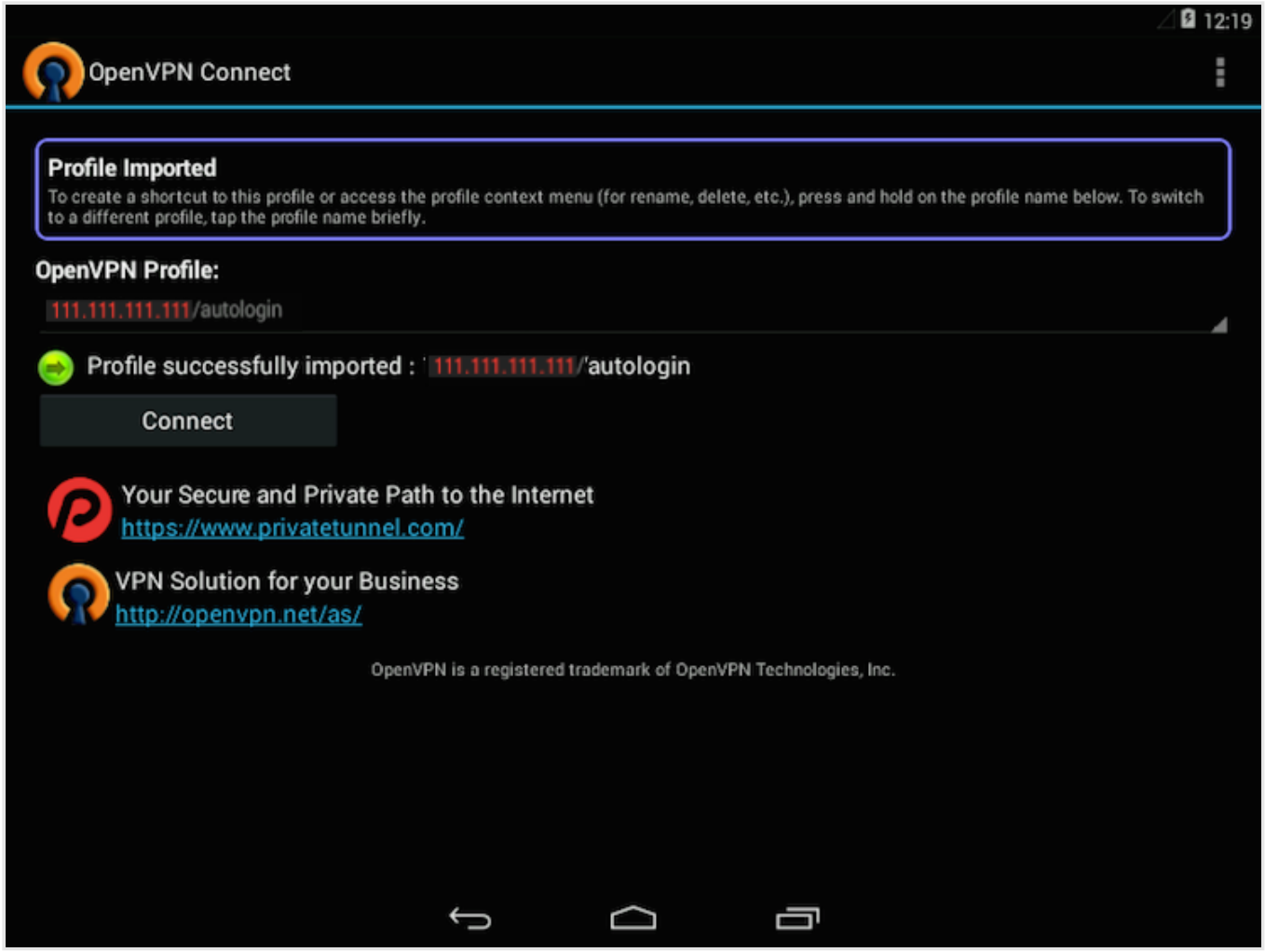


Depois, navegue até a localização do perfil salvo (a captura de tela utiliza `/sdcard/Download/`) e selecione o arquivo. O app fará uma notificação de que o perfil foi importado.



Conectando

Para conectar, simplesmente toque no botão **Connect**. Você será perguntado se você confia na aplicação OpenVPN. Escolha **OK** para iniciar a conexão. Para desconectar da VPN, volte para o app OpenVPN e escolha **Disconnect**.



Passo 13: Testar sua Conexão VPN

Depois de tudo instalado, um teste simples confirma que tudo está funcionando apropriadamente. Sem ter uma conexão VPN habilitada, abra um navegador e vá para DNSLeakTest.

O site irá retornar o endereço IP atribuído pelo seu provedor de Internet e como você aparece para o resto do mundo. Para verificar suas configurações DNS através do mesmo website, clique em **Extended Test** e ele irá lhe dizer quais servidores DNS você está usando.

Agora, conecte o cliente OpenVPN à VPN do seu Droplet e atualize o navegador. O endereço IP completamente diferente do seu servidor VPN deve aparecer agora. É assim que você aparece para o mundo agora. Novamente, DNSLeakTest's **Extended Test** irá checar suas configurações DNS e confirmar que você está agora usando os resolvedores DNS forçados pela sua VPN.

Passo 14: Revogando Certificados de Cliente

Ocasionalmente, você pode precisar revogar um certificado de cliente para prevenir futuros acessos ao servidor OpenVPN.

Para fazer isso, entre em seu diretório CA e recarregue o arquivo `vars`:

```
$ cd ~/openvpn-ca
$ source vars
```

Em seguida, chame o comando `revoke-full` usando o nome do cliente que você deseja revogar:

```
$ ./revoke-full client3
```

Isso irá mostrar alguma saída, terminando em `error 23`. Isso é normal e o processo deve ter gerado com êxito as informações de revogação necessárias, que são armazenadas em um arquivo chamado `crl.pem` dentro do subdiretório `keys`.

Transfira esse arquivo para o diretório de configuração `/etc/openvpn`:

```
$ sudo cp ~/openvpn-ca/keys/crl.pem /etc/openvpn
```

A seguir, abra o arquivo de configuração do servidor OpenVPN:

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

No final do arquivo, adicione a opção `crl-verify`, de forma que o servidor OpenVPN verifique a lista de revogação de certificado que criamos sempre que uma tentativa de conexão é feita:

/etc/openvpn/server.conf

```
crl-verify crl.pem
```

Salve e feche o arquivo.

Finalmente, reinicie o OpenVPN para implementar a revogação de certificados:

```
$ sudo systemctl restart openvpn@server
```

O cliente não deve mais ser capaz de se conectar com êxito ao servidor usando a credencial antiga.

Para revogar clientes adicionais, siga esse processo:

1. Gere uma nova lista de revogação de certificados recarregando o arquivo `vars` no diretório `~/openvpn-ca` e então chamando o script `revoke-full` com o nome do cliente.

2. Copie a nova lista de revogação de certificados para o diretório `/etc/openvpn` para sobrescrever a lista antiga.
3. Reinicie o serviço OpenVPN.

Esse processo pode ser utilizado para revogar quaisquer certificados que você tiver emitido previamente para seu servidor.

Conclusão

Parabéns! Agora você está atravessando com segurança a Internet protegendo sua identidade, localização, e tráfego dos bisbilhoteiros e censores.

Para configurar mais clientes, você precisa apenas seguir os passos **6**, e **11-13** para cada dispositivo adicional. Para revogar acesso de clientes, siga o passo **14**.

By: Justin Ellingwood

Translation:
Fernando Pimenta

Related Tutorials

Como Configurar sua Própria VPN com PPTP

How To Set Up an OpenVPN Server on Debian 9

How to Set Up an IKEv2 VPN Server with StrongSwan on Ubuntu 18.04

How To Set Up an OpenVPN Server on Ubuntu 18.04

Getting Started with Software-Defined Networking and Creating a VPN with ZeroTier One

---

12 Comments

Leave a comment...

Log In to Comment

washpereiraa *April 9, 2017*

0 Maravilhoso esse tutorial, queria ate dar um bjo e um abraço em quem fez.
Deu tudo certo até na parte de criar o script do cliente, disse que havia um erro de sintaxe '(' na linha 8, continuei o tutoria para ver o que resultava... e na hora de conectar o terminal local ao VPN pelo comando:

```
sudo openvpn --config client1.ovpn
```

ocorreu o seguinte erro:

```
Options error: In [CMD-LINE]:1: Error opening configuration file: client1.o
```

Use --help for more information.

Alguém pode me ajudar?

eeletro *May 12, 2017*

0 Olá, excelente tutorial!

Poderia dizer que mudanças poderiam ser feitas se eu utilizar o No-IP? Pois minha conexão é dinâmica.

Grato

josuemdsilva *June 5, 2017*

0 Cara obrigado pelo tutorial, segui a risca e esta conectando. Mas não consigo acessar a rede interna do local, conecto no servidor VPN e não acesso os servidores que estão na rede local. Tem algo mais em termos de firewall para configurar para acessar a rede local pelo openvpn ?

lionan *September 1, 2017*

0 comigo não resolve nome, porem eu acesso a rede interna toda pelo ip.
Experimenta o comando

ufw reset

coffeverton *October 17, 2017*

0 Pra mim também não resolve os hostnames, só acesso pelo ip.
Como resolver?

Callazzans *July 5, 2017*

0 Para que funcione corretamente no Android, é preciso comentar a linha "tls-auth ta.key 1".

Rickwacman *September 19, 2017*

0 de qual arquivo?

coffeverton *October 17, 2017*

0 Acredito que seja do base.conf, que fica na pasta ~/client-configs .
Pra mim já estava comentado.

Rickwacman *September 19, 2017*

Preciso autenticar via Active Directory. Como eu faria essa conexão? Existe algum tutorial?

0

---

fpproducoes *September 26, 2017*

0 Olá, muito obrigado pelo tutorial, muito bom mesmo!
Aqui para que eu conseguisse direcionar todo o fluxo tive que usar o seguinte tutorial:
https://community.openvpn.net/openvpn/wiki/NatHack

---

willianemanoel *January 11, 2018*

0 Muito bom o tutorial. A única coisa que não funcionou aqui foi navegar na internet através de um cliente Lonux.

Descomentei as linha indicadas e me certifiquei de que o script estava na pasta. Consigo pingar todos os outros clientes e o server. Só não consigo navegar na internet.

Alguma ideia?

---

brenouchoa *February 28, 2018*

0 Excelente tutorial! Só encontrei um errinho.
tem que alterar no conf o padrão do arquivo dh que esta com 1024 e no tutorial cria o 2048 bis.