EPx Home Page - OpenVPN HOWTO

OpenVPN HOWTO

OpenVPN HOWTO

Introdução

Este documento descreve os procedimentos para a configuração do OpenVPN em um ambiente típico de Escritório<->Casa. Enquanto esse HOWTO apresenta exemplos mais aprofundados de configuração, exemplos mais simples estão disponiveis na seção de exemplos da man page.

Este HOWTO também está disponivel em outros formatos.

- PDF (em inglês)
- PostScript (em inglês)

Existem 2 tipos basicos de túneis que você pode criar com o OpenVPN:

- **Tuneis IP Roteáveis** -- ideal para rotear túneis IP ponto-a-ponto sem broadcast. Ligeiramente mais eficientes que túneis ethernet em ponte e mais fáceis de se configurar. Esse HOWTO (abaixo) basea-se nesse tipo de tunel.
- **Tuneis Ethernet em Ponte** -- pode ser usado em túneis que rodam tanto IP quanto protocolos não IP. Esse tipo de tunel é apropriado para aplicações que rodam via broadcasts, como rede windows e jogos em LAN. Um pouco mais complexo de se implementar. Um Mini-HOWTO para túneis Ethernet em ponte (em inglês).

Outras documentações

Outras documentações e howtos estão disponiveis para a configuração do OpenVPN em diferentes configurações. (em inglês)

Tunel IP roteável HOWTO

Devido aos problemas envolvidos em configurar firewalls, VPNs, e NAT, nós iremos tentar descrever a implementação de uma solução completa, em vez de isolar a VPN apenas como um componente de um sistema maior.

Em nosso exemplo, tanto as redes internas da Casa e do Escritório estão ligadas a internet via 2 maquinas gateway que possuem um endereço IP publico e válido na internet. Cada maquina gateway contem 2 placas de rede, uma conectada a rede local e outra conectada a Internet. A maquina gateway serve NAT, firewall e serviços de VPN para as maquinas das suas respectivas redes internas. A configuração dos gateways tanto da Casa como do Escritório são similares, com a exeção que o escritório possui IP fixo, e a Casa possui IP dinamico fornecido via DHCP.

Todos os arquivos de configuração a seguir, estão disponiveis na distribuição do OpenVPN.

Parametros da rede IP para as redes de Casa e do Escritório

	Casa	Escritório
Sub-rede local (Endereço Privado)	10.0.1.0/24	10.0.0.0/24

Endereço do túnel (Endereço Privado)	10.1.0.2	10.1.0.1
Gateway OpenVPN (Endereço Público)	Cliente DHCP, sendo variável	1.2.3.4

Instalando o OpenVPN

Se o seu sistema não tem a biblioteca OpenSSL, você deve fazer o download e instala-la.

Se você quer obter os beneficios de compresão no Link VPN, ou quer instalar o OpenVPN com um pacote RPM, você deve instalar a biblioteca LZO.

Se você está usando o Linux com o Kernel versão 2.2 ou anterior, você deve instalar o Driver TUN/TAP. Usuários do kernel 2.4.7 ou superior já devem achar o driver junto com seu kernel. Usuários do kernel 2.4.0 até 2.4.6 devem ler aviso no final do arquivo INSTALL. (em inglês)

Agora faça o download do ultimo release do OpenVPN:
http://prdownloads.sourceforge.net/openvpn/openvpn-1.5.0.tar.gz

Instalação do tarball

Descompacte a distribuição:

```
gzip -dc openvpn-1.5.0.tar.gz | tar xvf
-
```

Compile o OpenVPN:

```
cd openvpn-1.5.0
./configure
make
make install
```

Se você não deseja as funcionalidades da biblioteca LZO, adicione o parametro **--disable-lzo** ao comando **./configure**. Outras opções podem ser ativadas, como o suporte a **pthread** (**./configure --enable-pthread**) para diminuir a latencia usando trocas de chaves SSL/TLS dinamicas. O comando

```
./configure --help
```

irá exibir todas as opções de configuração.

Instalação via RPM

Primeiramente crie o arquivo RPM. Isso exige que as bibliotecas OpenSSL, pthread, e LZO estejam presentes. Normalmente apenas a biblioteca LZO não está presente nas distruibuições de linux atuais.

```
rpmbuild -tb openvpn-1.5.0.tar.gz
```

O processo de criação do arquivo RPM, irá gerar muitas linhas de saida. Se o processo for bem sucedido, deverá existir uma linha próxima ao final da saida, iniciando com o nome do arquivo RPM criado. Agora você pode instalar o arquivo RPM binario com o comando:

```
rpm -Uvh binary-RPM-file
```

Configurando o driver TUN/TAP

Configurações a serem feitas uma unica vez

Se você está usando o kernel 2.4.7 ou superior, as chances são muito boas que seu kernel já tenha o driver TUN/TAP. Você pode confirmar isso com o comando

```
locate if_tun.h
```

que deve reportar um arquivo como **/usr/include/linux/if_tun.h**.

Para o kernel 2.4.7 ou superior, se você o instalou pelo tarball, entre com o seguinte comando para configurar o device TUN/TAP. (você pode omitir isso se você fez a instalação via RPM. Ele

já o fez automaticamente para você.):

```
mknod /dev/net/tun c 10 200
```

Se você está usando o kernel 2.2, você deve obter a Versão 1.1 do modulo TUN/TAP para o kernel e seguir as instruções da instalação.

Configurações a serem feitas a cada reboot

No linux, antes de usar o OpenVPN ou qualquer outro programa que use o driver TUN/TAP, você deve carregar o driver TUN/TAP

```
modprobe tun
```

e ativar o roteamento de pacotes IP:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Configurando o Firewall e o NAT

Esta seção assumen que você está usando o kernel 2.4 com um firewall baseado em **iptables**. Aqui está um exemplo de configuração do firewall que serve NAT para as maquinas na rede interna para o acesso a internet, stateful outgoing connection tracking, e suporte a OpenVPN:

sample-config-files/firewall.sh

```
#!/bin/bash

# Um exemplo de firewall amigavel ao OpenVPN.

# eth0 está conectada à internet.
# eth1 está conectada à rede interna.

# Troque a sub-rede para correponder a sua sub-rede interna.
# Casa irá usar 10.0.1.0/24 e o escritório irá usar 10.0.0.0/24.
PRIVATE=10.0.0.0/24

# Endereço loopback
LOOP=127.0.0.1

# Limpa todas as regras anteriores do iptables
# e bloqueia todo o tráfego temporariamente.
iptables -P OUTPUT DROP
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -F

# Seta as politicas padrão
iptables -P OUTPUT ACCEPT
iptables -P INPUT DROP
iptables -P FORWARD DROP

# Impede que pacotes destinados ao endereço loopback sejam recebidos pelas interfaces
iptables -A INPUT -i eth0 -s $LOOP -j DROP
iptables -A FORWARD -i eth0 -s $LOOP -j DROP
iptables -A INPUT -i eth0 -d $LOOP -j DROP
iptables -A FORWARD -i eth0 -d $LOOP -j DROP

# Qualquer coisa vinda da internet deve ter um endereço IP válido
iptables -A FORWARD -i eth0 -s 192.168.0.0/16 -j DROP
iptables -A FORWARD -i eth0 -s 172.16.0.0/12 -j DROP
iptables -A FORWARD -i eth0 -s 10.0.0.0/8 -j DROP
iptables -A INPUT -i eth0 -s 192.168.0.0/16 -j DROP
iptables -A INPUT -i eth0 -s 172.16.0.0/12 -j DROP
iptables -A INPUT -i eth0 -s 10.0.0.0/8 -j DROP

# Bloqueia tráfego NetBios de saída (se você tiver maquinas windows
# na rede interna). Isso não impede qualquer tipo de tráfego NetBios
# pelo túnel VPN, mas irá impedir que as maquinas windows
# da rede interna se comuniquem via NetBios pela internet
iptables -A FORWARD -p tcp --sport 137:139 -o eth0 -j DROP
iptables -A FORWARD -p udp --sport 137:139 -o eth0 -j DROP
iptables -A OUTPUT -p tcp --sport 137:139 -o eth0 -j DROP
iptables -A OUTPUT -p udp --sport 137:139 -o eth0 -j DROP

# Checa o endereço de origem dos pacotes que irão para a internet
iptables -A FORWARD -s ! $PRIVATE -i eth1 -j DROP

# Permite comunicação via loopback
iptables -A INPUT -s $LOOP -j ACCEPT
iptables -A INPUT -d $LOOP -j ACCEPT

# Permite que a maquina responda PINGs (pode ser desativado)
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

# Permite serviços como www e ssh (pode ser desativado)
```

```
iptables -A INPUT -p tcp --dport http -j ACCEPT
iptables -A INPUT -p tcp --dport ssh -j ACCEPT

# Permite que pacotes do OpenVPN entrem.
# Duplique a linha a seguir para cada
# cada tunel OpenVPN, trocando a opção --dport
# para a porta usada no tunel.
#
# No OpenVPN o número da porta é
# controlado pela opção --port n.
# Se você colocar essa opção no arquivo de
# configuração, você pode remover o '--'
#
# Se você está usando stateful firewall,
# pode comentar essa linha.

iptables -A INPUT -p udp --dport 5000 -j ACCEPT

# Permite que pacotes vindo de uma interface
# TUN/TAP entrem na rede.
# Quando o OpenVPN é iniciado em modo seguro,
# ele irá autenticar os pacotes antes de
# permitir a sua entrada na interface TUN/TAP.
# interface.  Portanto, não é necessario adicionar
# qualquer filtro aqui, a menos que você queira
# restringir o tráfego que pode passar pelo seu túnel.

iptables -A INPUT -i tun+ -j ACCEPT
iptables -A FORWARD -i tun+ -j ACCEPT
iptables -A INPUT -i tap+ -j ACCEPT
iptables -A FORWARD -i tap+ -j ACCEPT

# Permite pacotes originados da rede local.
iptables -A INPUT -i eth1 -j ACCEPT
iptables -A FORWARD -i eth1 -j ACCEPT

# Mentar o estado as conexões da maquina local e da rede interna
iptables -A OUTPUT -m state --state NEW -o eth0 -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state NEW -o eth0 -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Faz o NAT para a rede local
iptables -t nat -A POSTROUTING -s $PRIVATE -o eth0 -j MASQUERADE
```

OpenVPN oferece algumas opções adicionais na configuração do firewall:

- Se ambas as pontas do OpenVPN estiverem configuradas com o endereço IP da outra ponta com a opção **--remote**, e apenas firewalls stateful com connection tracking (como o **iptables**) existirem entre as pontas, é possivel rodar o OpenVPN sem qualquer tipo de regra de firewall expecifica, desde que ambas as pontas enviem pings para a outra ponta para manter a conexão ativa. Para fazer isso, simplesmente rode o OpenVPN com a opção **--remote *IP-remoto***, e **--ping 15** para ter certeza que pacotes irão passar pela VPN a cada 15 segundos.
- A opção acima é menos prática se uma das pontas troca seu endereço IP frequentemente, como um cliente DHCP ou um usuário de modem discado. Nesses casos, o exemplo de firewall acima, irá permitir que qualquer pacote UDP com destino a porta 5000 seja repassado ao OpenVPN. Isto pode ser considerado seguro em qualquer modo seguro do OpenVPN, já que todo tráfego pelo túnel deve passar por um teste de autenticação, ou ele será descartado.
- Se você escolher em deixar a porta do OpenVPN totalmente aberta, como no exemplo acima, você pode querer usar a opção **--tls-auth** para dobrar a autenticação no canal de controle TLS, usando tanto a chave RSA e uma senha pré-compartilhada como uma segunda linha de defesa contra ataques DoS. Para mais informações sobre a opção **--tls-auth**, veja a man page do openvpn.(em inglês)

Gerando os Cerificados e Chaves RSA

O OpenVPN tem 2 tipos de operação segura, uma baseada em SSL/TLS, usando certificados e chaves RSA, e outro usando uma chave estática pré-compartilhada. Enquanto que as chaves SSL/TLS + RSA são seguramente a opção mais segura, as chaves estaticas tem a vantagem de serem muito mais simples. Se você deseja usar chaves RSA, leia a seguir. Se você deseja usar chaves estáticas pré-compartilhadas, pule para a seção Gerando a chave estática Pré-Compartilhada.

Nós iremos gerar os certificados e chaves RSA usando o comando **openssl** incluso na biblioteca OpenSSL.

Os certificados RSA são chaves publicas que contém vários campos de segurança, como os campos **Common Name** ou **Email Address**. O OpenVPN tem a habilidade de fazer alguns testes para melhorar a segurança. Para mais informações, veja a opção **--tls-verify** na man page do Openvpn. (em inglês)

Em nosso exemplo iremos seguir a convenção do **apache** de usar a extenção **.crt** para os arquivos de certificados e a extenção **.key** para os arquivos de chaves privadas. Os arquivos de chaves privadas devem sempre serem mantidas seguras, já os arquivos de chaves privadas, podem ser livremente publicadas e distribuidas.

Selecione uma maquina, como a maquina do escritorio, para ser a maquina do gerenciamento de chaves.

Primeiro edite o arquivo **/usr/share/ssl/openssl.cnf** (ele pode estar em um lugar diferente, então use o comando **locate openssl.cnf** para encontra-lo).

Você deve fazer algumas modificações nesse arquivo:

- Criar um diretorio para servidor como um diretorio de trabalho, e mudar a opção **dir** apontando para ele.
- Considere aumentar o valor do parametro **default_days** ou sua VPN irá parar de funcionar misteriosamente após exatamente um ano.
- Aponte as opções **certificate** e **private_key** para o seu certificado mestre da sua autoridade certificadora e as chaves privadas que iremos gerar. Nos exemplos abaixo, nós iremos assumir que certificado mestre da sua autoridade certificadora se chama **my-ca.crt** e que a chave privada se chama **my-ca.key**.
- Crie os arquivos **index.txt** e **serial**. Crie o arquivo **index.txt** em branco e **serial** para conter um numero inicial serial, como **01**.
- Se você for paranóico sobre o tamanho de suas chaves, aumente a opção **default_bits** para 2048. O OpenVPN não irá ter nenhum problema em trabalhar com uma chave RSA de 2048 bits se você compilou o OpenVPN com suporte a biblioteca **pthread**, para ativar o suporte do processamento das chaves RSA em background. Você ainda pode usar chaves maiores mesmo sem o suporte a biblioteca **pthread**, mas você irá ver alguma degradação de performace no tunel durante as negociações das chaves SSL/TLS. Para um bom artigo sobre escolher um tamanho de chave RSA, veja a Edição de Abril de 2002 da *Bruce Schneier's Crypto-Gram Newsletter.*

Depois que o arquivo **openssl.cnf** foi editado, crie o par certificado/chave da sua entidade certificadora:

```
openssl req -nodes -new -x509 -keyout my-ca.key -out my-ca.crt -days 3650
```

Isto irá gerar o par certificado/chave-privada da sua entidade certificadora com validade de 10 anos.

Agora crie o par certificado/chave-privada tanto para Casa quanto para o Escritório:

```
openssl req -nodes -new -keyout escritorio.key -out escritorio.csr
openssl ca -out escritorio.crt -in escritorio.csr
openssl req -nodes -new -keyout casa.key -out casa.csr
openssl ca -out casa.crt -in casa.csr
```

Agora copie **casa.crt**, **casa.key**, e **my-ca.crt** para a maquina de casa sobre um canal seguro, embora apenas os arquivos **.key** são realmente considerados não-publicos.

Agora crie os parametros Diffie Hellman na maquina do escritório com o comando

```
openssl dhparam -out dh1024.pem 1024
```

Aumente o numero de bits de 1024 para 2048 se você também o aumentou no arquivo **openssl.cnf**.

Para os paranoicos, considere omitir a opção **-nodes** nos comandos **openssl** acima. Isso irá fazer com que cada chave privada seja criptografada com uma senha, fazendo com que suas chaves fiquem seguras se alguem acessar seu servidor e roube suas chaves. O problema dessa opção, é que você irá ter que digitar uma senha toda vez que você rodar o OpenVPN. Para mais informações, veja a opção **--askpass** na man page do openvpn. (em inglês)

Se você achar o gerenciamento manual de chaves RSA, note que o OpenVPN irá trabalhar com qualquer certificado X509 ou serviço incluindo as Autoridades Certificadoras comerciais como Thawte ou Verisign. Veja também o projeto OpenCA como um exemplo do que está sendo feito no mundo do Código Aberto para mudar essa realidade.

O OpenVPN também tem um pequeno conjunto de scripts que podem ser usadados para simplificar O gerenciamento de chaves e certificados RSA (em inglês).

Nota importante sobre o uso de Autoridades Certificadoras comerciais (CAs) com o OpenVPN

Você deve ter percebido que o modelo de seguraça do OpenVPN em modo SSL/TLS é orientado para os usuarios que irão gerar seu proprio certificado raiz e virar seu próprio CA. No modo SSL/TLS, o OpenVPN autentica a outra ponta checando se o certificado fornecido pela outra ponta é assinado pelo certificado da Autoridade Certificadora, que é expecificado com a opção **--ca**. Como no modelo de seguraça SSL da web, a segurança do OpenVPN em modo SSL/TLS está baseada na dificuldade de se forjar um certificado raiz.

Esse tipo de autenticação funciona muito bem se você gerou seu próprio certificado raiz, mas é um problema se você usa o certificado raiz de uma Autoridade Certificadora comercial como a Thawte. Se você por exemplo, colocou o certificado raiz da Thawte na opção **--ca**, qualquer certificado assinado pela Thawte será capaz de se autenticar com a sua ponta OpenVPN -- uma coisa que você certamente não quer.

Felizmente existe uma solução para este problema na opção **--tls-verify**. Essa opção irá permitir a execução de um comando para checar o conteudo do certificado, para selecionar qual certificado é permitido ou não. Veja o script **verify-cn** do diretorio *sample-scripts* para um exemplo de como fazer isso. Também consulte a man page na opção **--tls-verify**.

Arquivo de configuração usando o modo SSL/TLS e certificados/chaves RSA

Em nosso exemplo, nós iremos usar os arquivos de configuração do OpenVPN. O OpenVPN permite que as suas opções de configuração sejam passadas tanto pela linha de comando ou em um ou mais arquivos de configuração. As opções em arquivos de configurção podem omitir o "--" que é necessario para as opções em linha de comando.

Crie os seguintes arquivos de configuração:

sample-config-files/tls-office.conf

```
#
# Exemplo de arquivo de configuração do OpenVPN para o
# escritório usando modo SSL/TLS e certificados/chaves RSA.
#
# '#' ou ';' podem ser usados para delimitar comentarios.

# Usa um device dinamico tun.
# Para Linux Kernel 2.2 ou não linux,
# você deve usar um device numérico
# explicito como "tun1".
# OpenVPN também suporta device ethernet
# "tap" virtual.
dev tun

# 10.1.0.1 é o endereço local final da VPN (escritório).
# 10.1.0.2 é o endereço remoto final da VPN (casa).
ifconfig 10.1.0.1 10.1.0.2

# Esse script irá setar as rotas
# assim que a VPN for ativada.
up ./office.up
```

```
# Em uma troca de chaves SSL/TLS, Escritório irá
# assumir a parte do servidor, e a Casa
# irá assumir a parte do cliente.
tls-server

# Parametros Diffie-Hellman (apenas no tls-server)
dh dh1024.pem

# Certificado da Entidade Certificadora
ca my-ca.crt

# Nossa chave publica
cert office.crt

# Nossa chave privada
key office.key

# O OpenVPN usa por padrão a porta 5000 UDP.
# Cada tunel OpenVPN deve usar
# um numero de porta diferente.
# lport ou rport podem ser usados
# para configurar diferentes portas
# para local e remoto.
; port 5000

# Muda o UID e GID para
# "nobody" depois do inicio
# para uma segurança extra.
; user nobody
; group nobody

# Se você compilou o OpenVPN com
# a compactação LZO, descomente a
# linha a seguir.
; comp-lzo

# Envia um ping UDP para a outra ponta
# a cada 15 segundos para manter
# firewalls stateful connection
# ativos. Descomente isso se você usa
# um firewall stateful
; ping 15

# Descomente essa seção para uma detecção mais confiavel de quando
# um sistema perde sua conecxão. Por exemplo para modems, ou notebooks
# que viajam para outros lugares.
; ping 15
; ping-restart 45
; ping-timer-rem
; persist-tun
; persist-key

# Nivel de log.
# 0 -- silencioso, exeto para erros fatais.
# 1 -- quase silencioso, mas mostra erros de rede não fatais.
# 3 -- médio, ideal para uso diário.
# 9 -- barulhento, ideal para procura de erros.
verb 3
```

sample-config-files/office.up

```
#!/bin/bash
route add -net 10.0.1.0 netmask 255.255.255.0 gw $5
```

sample-config-files/tls-home.conf

```
#
# Exemplo de arquivo de configuração do OpenVPN para a
# casa usando modo SSL/TLS e certificados/chaves RSA.
#
# '#' ou ';' podem ser usados para delimitar comentarios.

# Usa um device dinamico tun.
# Para Linux Kernel 2.2 ou não linux,
# você deve usar um device numérico
# explicito como "tun1".
# OpenVPN também suporta device ethernet
# "tap" virtual.
dev tun

# A outra ponta OpenVPN é o gateway do escritório.
remote 1.2.3.4

# 10.1.0.2 é o endereço local final da VPN (casa).
# 10.1.0.1 é o endereço remoto final da VPN (escritório).
ifconfig 10.1.0.2 10.1.0.1

# Esse script irá setar as rotas
# assim que a VPN for ativada.
```

```
up ./home.up

# Em uma troca de chaves SSL/TLS, Escritório irá
# assumir a parte do servidor e a Casa
# irá assumir a parte do cliente.
tls-client

# Certificado da Entidade Certificadora
ca my-ca.crt

# Nossa chave publica
cert home.crt

# Nossa chave privada
key home.key

# OpenVPN usa por padrão a porta 5000 UDP.
# Cada tunel OpenVPN deve usar
# um numero de porta diferente.
# lport ou rport podem ser usados
# para configurar diferentes portas
# para local e remoto.
; port 5000

# Muda o UID e GID para
# "nobody" depois do inicio
# para uma segurança extra.
; user nobody
; group nobody

# Se você compilou o OpenVPN com
# a compactação LZO, descomente a
# linha a seguir.
; comp-lzo

# Envia um ping UDP para a outra ponta
# a cada 15 segundos para manter
# firewalls stateful connection
# ativos. Descomente isso se você usa
# um firewall stateful.
; ping 15

# Descomente essa seção para uma detecção mais confiavel de quando
# um sistema perde sua conecxão. Por exemplo para modems, ou notebooks
# que viajam para outros lugares.
; ping 15
; ping-restart 45
; ping-timer-rem
; persist-tun
; persist-key

# Nivel de log.
# 0 -- silencioso, exeto para erros fatais.
# 1 -- quase silencioso, mas mostra erros de rede não fatais.
# 3 -- médio, ideal para uso diário.
# 9 -- barulhento, ideal para procura de erros.
verb 3
```

sample-config-files/home.up

```
#!/bin/bash
route add -net 10.0.0.0 netmask 255.255.255.0 gw $5
```

Gerando a chave estática Pré-Compartilhada.

Em contraste com o gerenciamento de chaves RSA, o uso de uma chave pré-compartilhada estática tem o beneficio da simplicidade. Um dos problemas do uso de chave estática, é que por a chave nunca mudar, tudo o que passou pela VPN desde que ela foi ativa pode ser descriptografado sem maiores problemas, criando um furo de segurança muito grande.

Gere a chave estática com o seguinte comando:

```
openvpn --genkey --secret static.key
```

A chave estática é formatada em ascii e se parece como isso:

```
-----BEGIN OpenVPN Static key V1-----
e5e4d6af39289d53
171ecc237a8f996a
97743d146661405e
c724d5913c550a0c
30a48e52dfbeceb6
e2e7bd4a8357df78
4609fe35bbe99c32
bdf974952ade8fb9
```

```
71c204aaf4f256ba
eeda7aed4822ff98
fd66da2efa9bf8c5
e70996353e0f96a9
c94c9f9afb17637b
283da25cc99b37bf
6f7e15b38aedc3e8
e6adb40fca5c5463
-----END OpenVPN Static key V1-----
```

Uma chave estática do OpenVPN contem entropia o suficiente tanto para uma chave com uma cifra de 512 bits e uma chave de 512 bits HMAC para autenticação.

Copie **static.key** para a outra ponta por um canal seguro como **scp** ou copiar e colar via **ssh**.

Arquivo de configuração usando uma chave estática pré-compartilhada

Em nosso exemplo, nós iremos usar os arquivos de configuração do OpenVPN. O OpenVPN permite que as suas opções de configuração sejam passadas tanto pela linha de comando ou em um ou mais arquivos de configuração. As opções em arquivos de configurção podem omitir o "--" que é necessario para as opções em linha de comando.

Crie os seguintes arquivos de configuração:

sample-config-files/static-office.conf

```
#
# Exemplo de arquivo de configuração do OpenVPN para o
# escritório usando modo SSL/TLS e certificados/chaves RSA.
#
# '#' ou ';' podem ser usados para delimitar comentarios.

# Usa um device dinamico tun.
# Para Linux Kernel 2.2 ou não linux,
# você deve usar um device numérico
# explicito como "tun1".
# OpenVPN também suporta device ethernet
# "tap" virtual.
dev tun

# 10.1.0.1 é o endereço local final da VPN (escritório).
# 10.1.0.2 é o endereço remoto final da VPN (casa).
ifconfig 10.1.0.1 10.1.0.2

# Esse script irá setar as rotas
# assim que a VPN for ativada.
up ./office.up

# Nossa chave estática pré-compartilhada
secret static.key

# O OpenVPN usa por padrão a porta 5000 UDP.
# Cada tunel OpenVPN deve usar
# um numero de porta diferente.
# lport ou rport podem ser usados
# para configurar diferentes portas
# para local e remoto.
; port 5000

# Muda o UID e GID para
# "nobody" depois do inicio
# para uma segurança extra.
; user nobody
; group nobody

# Se você compilou o OpenVPN com
# a compactação LZO, descomente a
# linha a seguir.
; comp-lzo

# Envia um ping UDP para a outra ponta
# a cada 15 segundos para manter
# firewalls stateful connection
# ativos. Descomente isso se você usa
# um firewall stateful
; ping 15

# Descomente essa seção para uma detecção mais confiavel de quando
# um sistema perde sua conecxão. Por exemplo para modems, ou notebooks
# que viajam para outros lugares.
; ping 15
; ping-restart 45
; ping-timer-rem
; persist-tun
; persist-key
```

```
# Nivel de log.
# 0 -- silencioso, exeto para erros fatais.
# 1 -- quase silencioso, mas mostra erros de rede não fatais.
# 3 -- médio, ideal para uso diário.
# 9 -- barulhento, ideal para procura de erros.
verb 3
```

sample-config-files/office.up

```
#!/bin/bash
route add -net 10.0.1.0 netmask 255.255.255.0 gw $5
```

sample-config-files/static-home.conf

```
#
# Exemplo de arquivo de configuração do OpenVPN para a
# casa usando modo SSL/TLS e certificados/chaves RSA.
#
# '#' ou ';' podem ser usados para delimitar comentarios.

# Usa um device dinamico tun.
# Para Linux Kernel 2.2 ou não linux,
# você deve usar um device numérico
# explicito como "tun1".
# OpenVPN também suporta device ethernet
# "tap" virtual.
dev tun

# A outra ponta OpenVPN é o gateway do escritório.
remote 1.2.3.4

# 10.1.0.2 é o endereço local final da VPN (casa).
# 10.1.0.1 é o endereço remoto final da VPN (escritório).
ifconfig 10.1.0.2 10.1.0.1

# Esse script irá setar as rotas
# assim que a VPN for ativada.
up ./home.up

# Nossa chave estática pré-compartilhada
secret static.key

# O OpenVPN usa por padrão a porta 5000 UDP.
# Cada tunel OpenVPN deve usar
# um numero de porta diferente.
# lport ou rport podem ser usados
# para configurar diferentes portas
# para local e remoto.
; port 5000

# Muda o UID e GID para
# "nobody" depois do inicio
# para uma segurança extra.
; user nobody
; group nobody

# Se você compilou o OpenVPN com
# a compactação LZO, descomente a
# linha a seguir.
; comp-lzo

# Envia um ping UDP para a outra ponta
# a cada 15 segundos para manter
# firewalls stateful connection
# ativos. Descomente isso se você usa
# um firewall stateful
; ping 15

# Descomente essa seção para uma detecção mais confiavel de quando
# um sistema perde sua conecxão. Por exemplo para modems, ou notebooks
# que viajam para outros lugares.
; ping 15
; ping-restart 45
; ping-timer-rem
; persist-tun
; persist-key

# Nivel de log.
# 0 -- silencioso, exeto para erros fatais.
# 1 -- quase silencioso, mas mostra erros de rede não fatais.
# 3 -- médio, ideal para uso diário.
# 9 -- barulhento, ideal para procura de erros.
verb 3
```

sample-config-files/home.up

```
#!/bin/bash
route add -net 10.0.0.0 netmask 255.255.255.0 gw $5
```

Inciando a VPN em modo SSL/TLS

No gateway de Casa, inicie a VPN com o comando:

```
openvpn --config tls-home.conf
```

No escritorio, inicie a VPN com o comando:

```
openvpn --config tls-office.conf
```

Inciando a VPN com chaves estáticas

No gateway de Casa, inicie a VPN com o comando:

```
openvpn --config static-home.conf
```

No Escritorio, inicie a VPN com o comando:

```
openvpn --config static-office.conf
```

Teste a VPN

No gateway de Casa, teste a VPN pingando o Escritorio via VPN:

```
ping 10.1.0.1
```

No escritorio, teste a VPN pingado o gateway de casa via VPN:

```
ping 10.1.0.2
```

Se esses testes falarem silenciosamente, talvez você queira re-editar os arquivos de configura e aumentar o nivel de log para 8, que irá produzir informações muito mais detalhadas. Também consulte o FAQ para mais informações na resolução de problemas.

Se esses testes tiverem sucesso, agora tente pingar pelo tunel, usando as maquinas dentro das redes internas, para testar o roteamento. Basicamente qualquer maquina na sub-rede **10.0.1.0/24** deve ser capaz de se comunicar com qualquer maquina na sub-rede **10.0.0.0/24,** e vice e versa.

Se isso funcionar, parabéns! Se não, você pode querer checar os arquivos da mail list do OpenVPN e ver se alguem mais tem um problema similar ao seu. Se você não achar uma solução para seu problema aqui, considere posta-lo na lista **openvpn-users** list.

Fazendo a VPN compativel com DHCP

Em nosso exemplo acima, o gateway de casa tem um endereço IP dinamico que pode mudar sem qualquer aviso. Se você usa o dhcpcd como seu cliente dhcp, é facil contruir um script que pode ser executado qualquer hora que o endereço IP mudar. O script ficaria em algum lugar como **/etc/dhcpc/dhcpcd-eth0.exe**.

Basicamente, você deve adicionar uma linha neste script que iria mandar um sinal **SIGUSR1** ou **SIGHUP** para o daemon OpenVPN. Seria algo como:

```
killall -HUP openvpn
```

Quando o OpenVPN recebe este sinal, ele irá fechar e re-abrir a conecxão de rede com a outra ponta, usando o novo endereço IP designado pelo servidor DHCP.

Você também deve usar a opção **--float** se você está conectando a uma ponta que está sucetivel a mudanças de IP pelo servidor DHCP.

Também é possivel trabalhar com trocas de IP com o sinal **SIGUSR1** que é como o sinal **SIGHUP** exeto que ele oferece um controle melhor dos sub-sistemas do OpenVPN que ele reseta. O sinal SIGUSR1 também pode ser gerado internamente baseado nas opções **--ping** e **--ping-restart**. A opção **--persist-tun** permite um reset sem fechar e abrir o device TUN (que permite que conexões via vpn não se percam, mesmo durante uma troca de IP). A opção **--persist-remote-ip** permite a preservação do endereço IP remoto, mesmo durante uma troca de IP. Isso permite que ambas as pontas da VPN sejam clientes DHCP. A opção **--persist-key** não re-lê as chaves durante o restart (o que permite ao daemon OpenVPN ser reiniciado mesmo se seus privilégios forem diminuidos com **--user** ou **--group**).

Para mais informações sobre como usar o OpenVPN em um endereço dinamico, veja o FAQ.

OpenVPN também pode ser uando em casos onde ambas as pontas da VPN são dinamicas.

Inciando a VPN automaticamente no reboot

Primeiro crie um diretório para armazenar as chaves e arquivos de coniguração do OpenVPN, como **/etc/openvpn**.

Decida se você vai usar o OpenVPN no modo TLS ou chave estática, e copie os arquivos **.conf**, **.up**, **.key**, **.pem**, e **.crt** apropriados para o diretório **/etc/openvpn**.

Proteja seus arquivos **.key**:

```
chmod go-rwx /etc/openvpn/*.key
```

Se você está usando Linux com **iptables**, edite o arquivo de configuração do firewall, **firewall.sh**, fazendo as alterações apropriadas para seu sistema e o copie para o diretório **/etc/openvpn**.

Crie um script de inicialização que se pareça com isso:

sample-config-files/openvpn-startup.sh

```
#!/bin/bash

# Um exemplo de script de inicialição
# do OpenVPN para Linux.

# Diretorio de configurações do OpenVPN
dir=/etc/openvpn

# Carrega o firewall
$dir/firewall.sh

# Carrega o modulo TUN/TAP
modprobe tun

# ativa o roteamento de pacotes
echo 1 > /proc/sys/net/ipv4/ip_forward

# Chama o openvpn para cada tunel VPN
# em modo daemon. Você também pode remover
# a opção "--daemon" da linha de comando
# e adicionar a opção "daemon" ao arquivo
# de configuração.
#
# Cada tunel deve rodar em uma porta UDP
# separada. Use a opção "port" para controlar
# isso. Como todas as opções do OpenVPN, você
# pode especificar "--port 8000" na linha de
# comando, ou "port 8000" no arquivo de configuração.

openvpn --cd $dir --daemon --config vpn1.conf
openvpn --cd $dir --daemon --config vpn2.conf
openvpn --cd $dir --daemon --config vpn2.conf
```

E crie um script de desligamento como esse:

sample-config-files/openvpn-shutdown.sh

```
#!/bin/bash

# para com todos os processos do OpenVPN
```

```
killall -TERM openvpn
```

Finalmente, adicione chamadas para **openvpn-startup.sh** e **openvpn-shutdown.sh** para os scripts de inicio e desligamento de seu sistema, ou no diretório **/etc/init.d**.

Gerenciando o inicio e desligamento de varios túneis OpenVPN

Aqui está um examplo de um script do **/etc/init.d** que irá automaticamente iniciar um processo do OpenVPN para cada arquivo **.conf** que for encontrado no diretório **/etc/openvpn**.

Esse script é instalado por padrão quando você instala o OpenVPN via um pacote RPM. (N.T: Devido a problemas com quebra de linha, é recomendavel que você copie esse script da distribuição do OpenVPN)

sample-scripts/openvpn.init

```
#!/bin/sh
#
# openvpn       Este shell script cuida do inicio e finalização
#               do openvpn no RedHat ou em outro sistema baseado
#               no chkconfig.
#
# chkconfig: 345 80 30
#
# description: OpenVPN é um programa robusto e flexivel para tunelamento que
#              usa toda as funcionalidades de encriptação, autenticação e certificação
#              que a biblioteca OpenSSL fornece, para tunelar seguramente redes IP atraves
#              de uma simples porta UDP.
#

# Contribuido para o projeto OpenVPN por
# Douglas Keller <doug@voidstar.dyndns.org>
# 2002.05.15

# Para instalar:
#   copie este arquivo para /etc/rc.d/init.d/openvpn
#   shell> chkconfig --add openvpn
#   shell> mkdir /etc/openvpn
#   crie .conf ou .sh files em /etc/openvpn (veja abaixo)

# Para desinstalar:
#   execute: chkconfig --del openvpn

# Notas do autor:
#
# Eu criei um init script em /etc/init.d e alterei o openvpn.spec para automaticamente
# registrar o script. Uma vez que o RPM está instalado você pode iniciar e parar
# o OpenVPN com "service openvpn start" e "service openvpn stop".
#
# Esse init script faz o seguinte:
#
# - Inicia um processo OpenVPN para cada arquivo .conf que ele encontra em /etc/openvpn
#
# - Se /etc/openvpn/xxx.sh existe para um arquivo xxx.conf então ele o executa
#   antes de iniciar o OpenVPN (util para fazer openvpn --mktun...).
#
# - Alem de start/stop você também pode:
#
#   service openvpn reload - SIGHUP
#   service openvpn reopen - SIGUSR1
#   service openvpn status - SIGUSR2

# Modificações 2003.05.02
#   * Alterado '==' para '=' para compatibilidade com sh (Bishop Clark).
#   * Se condrestart|reload|reopen|status, checa se já não está iniciado (James Yonan).
#   * Adicionado lock, piddir, e work variaveis (James Yonan).
#   * Se o start é tentado 2 vezes, sem a intervenção de um stop, ou
#     se start é tentado quando o start anterior não foi bem sucessedido
#     ele mata qualquer outro processo antes de começar uma nova operação (James Yonan).
#   * Faz um trabalho melhor de detecção de erros no start, retornando sucesso ou falha (James Yonan).

# Localização do binário do OpenVPN
openvpn="/usr/sbin/openvpn"

# Arquivo de trava
lock="/var/lock/subsys/openvpn"

# Diretorio de PID
piddir="/var/run/openvpn"

# Diretório de trabalho
```

```
work=/etc/openvpn

# Código da biblioteca de função.
. /etc/rc.d/init.d/functions

# Código da configuração de rede.
. /etc/sysconfig/network

# Checa se a rede está ativa.
[ ${NETWORKING} = "no" ] && exit 0

[ -f  $openvpn ] || exit 0

# Ve como nós fomos chamados.
case "$1" in
  start)
        echo -n $"Starting openvpn: "

        /sbin/modprobe tun >/dev/null 2>&1

        # A partir de uma perspectiva segura, eu acho que faz
        # sentido remover isso, e forçar os usuarios que precisam disso
        # a ativar explicitamente em seus scripts --up ou nas configurações
        # de firewall.

        #echo 1 > /proc/sys/net/ipv4/ip_forward

        if [ ! -d  $piddir ]; then
            mkdir $piddir
        fi

        if [ -f $lock ]; then
            # nós não fomos desligados corretamente
            for pidf in /bin/ls $piddir/*.pid 2>/dev/null; do
              if [ -s $pidf ]; then
                kill at $pidf >/dev/null 2>&1
              fi
              rm -f $pidf
            done
            rm -f $lock
            sleep 2
        fi

        rm -f $piddir/*.pid
        cd $work

        # Inicia cada .conf em $work e executa .sh se existe
        errors=0
        successes=0
        for c in /bin/ls *.conf 2>/dev/null; do
            bn=${c%%.conf}
            if [ -f "$bn.sh" ]; then
                . $bn.sh
            fi
            rm -f $piddir/$bn.pid
            $openvpn --daemon --writepid $piddir/$bn.pid --config $c --cd $work
            if [ $? = 0 ]; then
                successes=1
            else
                errors=1
            fi
        done

        if [ $errors = 1 ]; then
            failure; echo
        else
            success; echo
        fi

        if [ $successes = 1 ]; then
            touch $lock
        fi
        ;;
  stop)
        echo -n $"Shutting down openvpn: "
        for pidf in /bin/ls $piddir/*.pid 2>/dev/null; do
          if [ -s $pidf ]; then
            kill at $pidf >/dev/null 2>&1
          fi
          rm -f $pidf
        done
        success; echo
        rm -f $lock
        ;;
  restart)
        $0 stop
        sleep 2
        $0 start
        ;;
  reload)
        if [ -f $lock ]; then
            for pidf in /bin/ls $piddir/*.pid 2>/dev/null; do
                if [ -s $pidf ]; then
                    kill -HUP at $pidf >/dev/null 2>&1
                fi
            done
        else
            echo "openvpn: service not started"
            exit 1
        fi
```

```
        ;;
    reopen)
        if [ -f $lock ]; then
            for pidf in /bin/ls $piddir/*.pid 2>/dev/null; do
                if [ -s $pidf ]; then
                    kill -USR1 at $pidf >/dev/null 2>&1
                fi
            done
        else
            echo "openvpn: service not started"
            exit 1
        fi
        ;;
    condrestart)
        if [ -f $lock ]; then
            $0 stop
            # avoid race
            sleep 2
            $0 start
        fi
        ;;
    status)
        if [ -f $lock ]; then
            for pidf in /bin/ls $piddir/*.pid 2>/dev/null; do
                if [ -s $pidf ]; then
                    kill -USR2 at $pidf >/dev/null 2>&1
                fi
            done
            echo "Status written to /var/log/messages"
        else
            echo "openvpn: service not started"
            exit 1
        fi
        ;;
    *)
        echo "Usage: openvpn
{start|stop|restart|condrestart|reload|reopen|status}"
        exit 1
esac

exit 0
```

Inicialização do OpenVPN usando inetd ou xinetd

O serviço **xinetd** pode ser usado para iniciar automaticamente o daemon do OpenVPN, assim que ele receber um datagrama UDP de uma ponta remota.

Essa configuração do xinetd irá fazer com que o xinetd escute a porta 5000 UDP esperando pelo primeiro datagrama de uma nova conexão OpenVPN (usando uma chave pré-compartilhada), quando então ele irá chamar o daemon do OpenVPN que irá conduzir a conexão. Note que o uso da chave **--inactive** irá fazer com que o daemon OpenVPN se desligue após 10 minutos de tempo inativo. Depois que o daemon OpenVPN se desligue por qualquer motivo, o serviço xinted irá escutar novamente a porta esperando pelo primeiro datagrama de uma nova conexão. Também note que o xinetd irá iniciar o daemon OpenVPN com privilégios de root, mas o OpenVPN irá (após ler os arquivos de chaves protegidos) descartar esses privilégios usando o usuario *nobody*.

O arquivo com a chave pode ser gerado com o seguinte comando:

```
openvpn --genkey --secret key
```

Note que cada túnel OpenVPN precisa rodar em um número de porta diferente, e precisa do seu próprio arquivo de configuração xinetd. Isso é necessario pois o OpenVPN precisa de informações especificas de cada nova conexão VPN em pontencial, incluindo chaves, interfaces TUN/TAP, endereços IP e informação de roteamento. Nesse momento no desenvolvimento do OpenVPN, ele ainda não é capaz de trabalhar com algum tipo de perfil de conexão que permitiria um simples arquivo de configução com varios clientes em pontencial. Como o OpenVPN é implementado via UDP, ele não pode levar vantagem da estrutura disponivel para criar um novo processo a cada nova conexão, como os disponiveis para servidores TCP, que escutam em uma porta fixa e depois trocam para outra porta dinamicamente. Entretanto essa funcionalidade será implantada quando houver interesse e suporte o suficiente da comunidade de desenvolvedores e de usuários.

sample-config-files/xinetd-server-config

```
# Um arquivo de configuração do xinetd para o OpenVPN.
#
# Esse arquivo deve ser renomeado para openvpn ou algo igualmente
# descritivo e copiar ele para o diretório /etc/xinetd.d.
# xinetd deve ser informado desse arquivo sendo reiniciado
# ou enviando a ele um sinal SIGHUP.
```

```
#
# Para cada cliente em potencial, você deve criar uma cópia desse
# arquivo de configuração em uma porta diferente. Também note que
# cada arquivo de chave e endereços IP devem ser únicos para cada
# cliente. Essa configuração assume que o executavel e as chaves do
# OpenVPN estão em /root/openvpn. Mude isso para a sua realidade.

service openvpn_1
{
        type            = UNLISTED
        port            = 5000
        socket_type     = dgram
        protocol        = udp
        wait            = yes
        user            = root
        server          = /root/openvpn/openvpn
        server_args     = --inetd --dev tun --ifconfig 10.4.0.2 10.4.0.1
--secret /root/openvpn/key --inactive 600 --user nobody
}
```

sample-config-files/xinetd-client-config

```
# Esse arquivo de configuração
# é a parte cliente do
# xinetd-server-config

dev tun
ifconfig 10.4.0.1 10.4.0.2
remote my-server
port 5000
user nobody
secret /root/openvpn/key
inactive 600
```

Copyright © 2002-2004 por James Yonan. Este documento é distribuido sobre a GNU Free Documentation License Version 1.2. Traduzido por Luiz Antonio Cassetari Vieira Filho.

Ir para índice

Copyright © 2006 Elvis Pfützenreuter. All rights reserved.

Comodo SSL - $15
Trusted by 99% of Browser. No more paperwork. SSL issued in 3 minutes

Maximize sua Banda Larga
Com uma linha telefônica VoIP. Faça o test-drive grátis do Vono

Navegue mais rápido. Baixe Firefox com Google Toolbar.