OpenVPN em Linux

Autor: Paulo Nave <pilao51 at hotmail.com>
Data: 01/07/2006

Início

Bem, vamos começar com o download dos softwares necessários, no caso aqui o *openVPN* em:

- http://openvpn.net/download.html

E também o *lzo* em:

- http://www.oberhumer.com/opensource/lzo/download/

Este segundo é opcional, pois podemos desabilitá-lo na hora em que instalarmos o openVPN.

Agora que já baixamos os pacotes, vamos partir para a instalação em si.

1. Vamos descompactar os pacotes, que estão como .tar.gz, primeiro o openVPN:

 # tar -zxvf openvpn-2.0.7.tar.gz -C /usr/local/src

Aqui peço pra ele fazer a descompactação no diretório /usr/local/src, por opção própria vocês podem fazer isso onde quiserem. Agora o lzo:

 # tar -zxvf lzo-2.02.tar.gz -C /usr/local/src

Agora que já descompactamos, vamos para compilação em si, todos sabem que cada distribuição precisa de pacotes extras para se compilar um programa, que na maioria das vezes devem ser instalados depois que a instalação do *Linux* for terminada, neste caso como estamos utilizando o *Conectiva 10*, vamos instalar os seguintes pacotes:

 # apt-get install task-kernel-compiling
 # apt-get install glibc
 # apt-get install glibc-devel
 # apt-get install g77
 # apt-get install c++

Em outras distribuições vocês podem usar os respectivos programas de gerenciamento de pacote, como *DrakConf* - este em modo gráfico ou o próprio *apt-get* no Debian.

Compilando e instalando o openVPN

Agora que já temos os pacotes descompactados e os programas necessários para a compilação, vamos para a mesma.

Primeiro precisaremos de um pacote chamado *openSSL*, na maioria das vezes ele já foi instalado na hora em que vocês instalaram o *Linux*, mas se não, a maioria das distribuições tem ele nos CDs de instalação. Aqui no *Conectiva* 10 eu fiz a instalação padrão e ele veio junto instalado, mas se por algum acaso vocês não tiverem ele, instalem pelo gerenciador de pacote das respectivas distribuições, aqui usaremos se necessário:

 # apt-get openssl
 # apt-get openssl-devel
 # apt-get openssl-progs

Estes dois últimos acredito que vocês precisaram instalar, isso no Conectiva 10.

Agora entre no diretório em que descompactamos os arquivos, no caso o /usr/local/src e dê um "ls", note que temos dois diretórios, o do *lzo* e do *openVPN*. Primeiro vamos entramos no do lzo e digitar o seguinte:

 # ./configure

Este arquivo configura os parâmetros para a instalação ou compilação em si.

 # make

Este comando compila o programa.

 # make install

e finalmente este instala o programa.

Sempre que vocês tiverem dúvidas a respeito da instalação, procurem na pasta dos programas descompactados pelo arquivo INSTALL, nele estão contidas as formas de instalação do mesmo, é bem útil sempre dar uma olhada.

Agora vamos para o openVPN, entre na pasta dele, que no caso também está em /usr/local/src e digite:

 # ./configure

Neste primeiro nós podemos usar uma segunda opção, para quem não quis instalar o lzo que ficaria assim:

 # ./configure --disable-lzo

Mas se vocês instalaram o lzo como acima, mantém-se somente o "./configure". Depois:

 # make
 # make install

Pronto, se correu tudo bem já temos o OpenVPN instalado, agora vamos configurar o mesmo.

Configurando o OpenSSL

Agora chegamos em uma parte um pouco mais complicada, mas também nenhum bicho de sete cabeças.

Primeiro vamos lidar com a segurança, por isso o pacote *openSSL*, ele cria certificados de segurança, como os usados em páginas de banco para aumentar a segurança dos dados transferidos de uma máquina para outra.

Então vamos lá, encontre o arquivo *openssl.cnf*, neste exemplo ele está dentro de /etc/ssl, mas se precisar localizá-lo digite:

 # find / -name openssl.cnf

Agora vamos editá-lo:

 # vi openssl.cnf

Dentro dele procure por [CA_defaults] - fica logo no começo do arquivo e altere as respectiva linha como no exemplo:

```
dir = /etc/ssl/certificados # Where everything is kept

certificate = $dir/my-ca.crt # The CA certificate
private_key = $dir/my-ca.key # The private key
```

Saia do arquivo e salve com o comando :wq.

Agora dentro do diretório /etc/ssl, crie o diretório "certificados" com o comando:

 # mkdir certificados

Acesse o mesmo:

 # cd certificados

e digite:

 # touch index.txt
 # echo 01 > serial

Ótimo, agora vamos criar o certificado e a chave ainda dentro do diretório /etc/ssl/certificados, digite:

 # openssl req -nodes -new -x509 -keyout my-ca.key -out my-ca.crt -days 365
Generating a 1024 bit RSA private key
.....+++++++
....+++++++
writing new private key to 'my-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) [AU]:BR --> Preencha os campos todos com letra maiuscula
State or Province Name (full name) [Some-State]:SP
Locality Name (eg, city) []:JAÚ
Organization Name (eg, company) [Internet Widgits Pty Ltd]:INET
Organizational Unit Name (eg, section) []:VPN

Common Name (eg, YOUR name) []:PAULO
Email Address []:PILAO51@HOTMAIL.COM

Feito isso, vamos executar mais um comando, sempre dentro do diretório /etc/ssl/certificados. Para terminar a criação do certificados e chaves:

 # openssl dhparam -out dh.pem 1024

Pronto! Fase completa.

Configurar OpenVPN

Segunda fase, agora vamos configurar o *OpenVPN* propriamente dito. Ainda dentro do diretório /etc/ssl/certificados, crie mais um subdiretório.

 # mkdir newcerts

Agora volte até o prompt de comando raiz com o comando:

 # cd /

e crie o diretório openvpn dentro de /etc:

 # mkdir /etc/openvpn
 # chmod 700 /etc/openvpn
 # ln -s /etc/ssl/certificados /etc/openvpn

Feito isso voltamos para o diretório /etc/ssl/certificados e vamos criar nosso certificado público e privado para a VPN. Aqui vou chamá-los de pilao51, mas cada um pode chamar do que quiser. Executamos então dois comandos:

 # openssl req -nodes -new -keyout pilao51.key -out pilao51.csr

Este primeiro fará basicamente as mesmas perguntas do que usamos pra criar o certificado e chave anteriormente. Responda da mesma forma com letras maiúsculas menos o campo password, esse vocês podem deixar em branco, depois execute:

 # openssl ca -out pilao51.crt -in pilao51.csr
Using configuration from /etc/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
Serial Number: 1 (0x1)
Validity
Not Before: Oct 31 00:30:24 2003 GMT
Not After : Oct 30 00:30:24 2006 GMT
Subject:
countryName = BR
stateOrProvinceName = SP
organizationName = INET
organizationalUnitName = VPN
commonName = PAULO
emailAddress = PILAO51@HOTMAIL.COM
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
F1:4A:6C:C7:7D:EB:FD:13:BE:2B:6B:16:91:98:AE:AD:BA:DE:E4:02
X509v3 Authority Key Identifier:
keyid:D7:3B:97:76:AE:AE:E2:C8:5D:46:D7:58:93:4D:31:C1:71:65:36:C5
DirName:/C=BR/SP=SAO PAULO/L=JAU/O=INET /OU=VPN/CN=PAULO/emailAddress=PILAO51@HOTMAIL.COM
serial:00
Certificate is to be certified until Oct 30 00:30:24 2006 GMT (1095 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

Responda as perguntas sempre com "y".

Terminado isto vocês terão criados 3 arquivos: pilao51.crt, pilao51.csr e pilao51.key.

Agora iremos criar o arquivo de configuração que servirá tanto para o servidor quanto para o cliente.

Configurar OpenVPN - parte 2

Acesse o diretório /etc/openvpn e dê o comando "ls", você vai notar que já existe um link simbólico que nós mesmos criamos anteriormente com o nome de "certificados", então ótimo, agora vamos criar o arquivo de configuração com o comando:

 # vi pilao51.conf

NOTA: O nome aqui será o mesmo que você usou para criar os certificados e chaves anteriormente. Com o arquivo aberto, digite como abaixo:

```
# Exemplo de configuração do OpenVPN
# para o servidor usando o modo SSL/TLS e chaves RSA.
#
# '#' ou ';' são comentários.
#
# OBS: Tradução livre do arquivo
# sample-config-files/tls-office.conf
# no diretório de sources do OpenVPN.

# Usar como interface o driver tun.
dev tun

# 10.10.0.1 é o nosso IP local (Servidor).
# 10.10.0.2 é o IP remoto (Cliente).
ifconfig 10.10.0.1 10.10.0.2
# Esses IPs não precisam ser exatamente os mesmos da rede
# interna como 192.198.0.1.ou 192.168.1.10
# É até melhor manter-se esses mesmos

# Diretório onde estão todas as configurações
cd /etc/openvpn

# O OpenVPN irá executar esse script
# quando o túnel estiver carregado.
# Ideal para setar as rotas
up ./pilao51.up

# No modo SSL/TLS a matriz irá
# assumir a parte do servidor,
# e a filial será o cliente.
tls-server

# Parâmetros Diffie-Hellman (apenas no servidor)
dh dh.pem

# Certificado da CA
ca my-ca.crt

# Certificado público da Matriz
cert pilao51.crt

# Certificado privado da Matriz
key pilao51.key

# OpenVPN usa a porta 5000/UDP por padrão.
# Cada túnel do OpenVPN deve usar
# uma porta diferente.
# O padrão é a porta 5000
; port 5000

# Mudar UID e GID para
# "nobody" depois de iniciado
# para uma segurança extra.
; user nobody
; group nobody

# Envia um ping via UDP para a parte
# remota a cada 15 segundos para manter
# a conexão em firewall statefull
# Muito recomendado, mesmo se você não usa
# um firewall baseado em statefull.
ping 15

# Nível dos logs.
# 0 -- silencioso, exceto por erros fatais.
# 1 -- quase silencioso, mas mostra erros não fatais da rede.
# 3 -- médio, ideal para uso no dia-a-dia
# 9 -- barulhento, ideal para solução de problemas
verb 3
```

Salve o mesmo com o comando :wq - Note como explicado acima que os IPs 10.10.0.1 e 10.10.0.2 são virtuais, específicos para a VPN.

Agora crie o arquivo "pilao51.up", ele está constando no nosso arquivo de configuração e serve para especificar as rotas do IP. Claro, crie ele dentro de /etc/openvpn.

 # vi pilao51.up

E digite os comandos abaixo:

```
#!/bin/bash
route add -net 192.168.0.0 netmask 255.255.255.0 gw $5
```

Salve-o com o comando :wq - Note que esse contém o IP da rede interna com a classe aqui definida como 192.168.0.0 para IPs de 192.168.0.1 a 192.168.0.254. Depois de salvo, execute o comando:

 # chmod +x pilao51.up

Isso vai torná-lo executável.

Ótimo, já temos os arquivo para o servidor pronto, é servidor pois é ele que vai ceder o número IP WAN ou da internet, que pode ser ADSL speedy, A cabo, etc.

Agora então executamos o *openVPN*. Quando o compilamos ele não cria um daemon de inicialização, como aqueles com {start:stop:restart:status}, então vamos iniciá-lo com o comando abaixo:

OBS: No final deste tutorial passarei um script que contém as instruções para se tentar iniciar o openVPN da maneira start e stop.

 # /usr/local/sbin/openvpn --config /etc/openvpn/pilao51.conf --daemon

Se correr tudo bem e voltar para o prompt sem nenhuma mensagem, daí então executamos um "tail /var/log/messages" para ver o que aconteceu, é para aparecer mais ou menos isto:

OpenVPN 1.4.3 i686-pc-linux-gnu [SSL] built on Oct 13 2003
UDP link local (bound): [undef]:5000
UDP link remote: [undef]
Diffie-Hellman initialized with 1024 bit key
Data Channel MTU parms [MUITOS PARAMETROS]
Control Channel MTU parms [MUITOS PARAMETROS]
TUN/TAP device tun0 opened
/sbin/ifconfig tun0 10.1.0.1 pointopoint 10.1.0.2 mtu 1259
/pilao51.up tun0 1259 1300 10.10.0.1 10.10.0.2

Com isto o servidor já está pronto para receber o cliente.

Para parar o serviço, execute "ps ax" e procure pela linha:

/usr/local/sbin/openvpn --config /etc/openvpn/pilao51.conf --daemon

veja seu pid e execute:

 # kill [número do pid]

Configurar cliente Windows

Já estamos quase no fim, nesta fase começaremos a configurar o cliente, que no caso será o Windows XP. Também há como configurar clientes *Linux*, mas não abordarei este tipo de configuração neste artigo. Quem precisar deste tipo de configuração pode acessar a pagina:

- http://www.altoriopreto.com.br/artigos_3rd/artigo_vpn.php

Que foi onde me baseei para montar parte deste artigo.

Mas voltando, antes de irmos para o próprio Windows, precisaremos de alguns arquivos que ainda estão no Linux. Acesse o diretório /etc/ssl/certificados e copie para um disquete os seguintes arquivos:

my-ca.key - pilao51.crt - pilao51.key - dh.pem - my-ca.crt

Depois ainda no diretório /etc/ssl/certificados, crie o arquivo "pilao51.ovpn", lembre-se esse arquivo tem o mesmo nome dos outros que você criou, como o .conf e .up, aqui no caso é pilao51.

 # vi pilao51.ovpn

e copie como abaixo:

```
# Exemplo de configuração do OpenVPN
# para o cliente, usando o modo SSL/TLS e chaves RSA.
#
# '#' ou ';' são comentários.
#
# OBS: Tradução livre do arquivo
# sample-config-files/tls-office.conf
# no diretório de sources do OpenVPN.

# Usar como interface o driver tun.
dev tun

# IP da parte remota.
remote 200.xxx.xxx.xxx.

# Aqui é o pulo do gato, você deve trocar
# esse endereço IP pelo endereço da internet do servidor

# 10.10.0.2 é o nosso IP local (cliente)
# 10.10.0.1 é o IP remoto (servidor)
ifconfig 10.10.0.2 10.10.0.1
# Note que aqui invertemos os endereços virtuais
# O primeiro passou a ser o do cliente
# Depois o do servidor

# Ao invés de rodar um script para setar as rotas,
# vamos setar no arquivo de configuração.
route 192.168.1.0 255.0.0.0 10.10.0.1
# E aqui o esquema é o mesmo do IP da
# rede local, no caso 192.168.1.0 é a classe com IPs de
# 192.168.1.1 a 192.168.1.254
# apontando para o IP remoto do servidor

# No modo SSL/TLS a matriz irá
# assumir a parte do servidor
# e a filial será o cliente.
tls-client

# Certificado da CA
ca my-ca.crt

# Certificado público da Filial
cert pilao51.crt

# Certificado privado da Filial
key pilao51.key

# OpenVPN usa a porta 5000/UDP por padrão.
# Cada túnel do OpenVPN deve usar
# uma porta diferente.
# O padrão é a porta 5000
port 5001

# Envia um ping via UDP para a parte
# remota a cada 15 segundos para manter
# a conexão em firewall statefull
# Muito recomendado, mesmo se você não usa
# um firewall baseado em statefull.
ping 15

# Nível dos logs.
# 0 -- silencioso, exceto por erros fatais.
# 1 -- quase silencioso, mas mostra erros não fatais da rede.
# 3 -- médio, ideal para uso no dia-a-dia
# 9 -- barulhento, ideal para solução de problemas
verb 3
```

Feito isso, salve com o comando :wq e também salve no disquete com os outros arquivos, eles serão levados para o cliente Windows.

OBS: Leia todo o arquivo de configuração, isto é muito importante. Tanto este como o do servidor, você vai notar que há várias partes dele com o pilao51 inserido, você poderá ter que mudar isto para o nome que escolheu.

Configurar cliente Windows - parte 2

Ótimo, estamos quase no fim mesmo. Agora vamos pegar o disquete com os arquivos e levar até o cliente Windows, mas primeiro vamos precisar instalar o *OpenVPN* no Windows XP. Para isso acessamos o endereço:

- http://openvpn.net/download.html

e fazemos o download do mesmo para Windows, normalmente é o nome do arquivo .exe, feito isto vamos instalar.

Depois de feita a instalação, copie todos os arquivos que estão no disquete para o diretório:

C:Arquivos de ProgramasOpenVPNconfig

Normalmente é aí que o OpenVPN se instala, feito isto abra um terminal do DOS, mais conhecido como Prompt de comando, ele sempre fica em:

Iniciar---> Acessórios
ou
Iniciar---> Executar--->command.

Agora vá até o diretório C:Arquivos de ProgramasOpenVPNconfig com o comando:

cd Arquivos de ProgramasOpenVPNconfig

E dê um "dir" para verificar se os arquivos trazidos com o disquete estão aí mesmo. Confirmado execute ainda no prompt:

openvpn --config pilao51.ovpn
OpenVPN 1.5_beta13 Win32-MinGW [SSL] [LZO] built on
Control Channel MTU parms [L:1541 D:138 EF:38 EB:0ET:0]
TAP-WIN32 device [ConexÒo de rede local 4] opened:
.{72015FC0-61D0-46F8-9265-E5D0B9EAF16B}.tap
CORTA.....
CORTA.....
CORTA.....
Data Channel Decrypt: Using 160 bit message hash '
HMAC authentication
Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA
AES256-SHA, 2048 bit RSA
Peer Connection Initiated with 200.0.0.1:5001

Pronto, conseguimos nos conectar. Agora dê um "ping 10.10.0.1", se pingar ótimo, está no ar. Parabéns!

Conclusão

Com nossa VPN no ar irão aparecer outras dúvidas, mas deixo pra vocês agora os detalhes para a melhor forma de ajeitar a mesma.

Esse artigo foi montado com base em outros 2:

OpenVPN
Por Luiz Antonio Cassetari Vieira Filho
http://www.altoriopreto.com.br/artigos_3rd/artigo_vpn.php
OpenVPN no Debian (Stable) com cliente Windows 2000 ou XP
Por Kairo Araújo
http://www.kairo.eti.br/linux-notes-old/openvpn-client_win2k-xp.html

Só tenho a agradecer estas duas pessoas pelos excelentes artigos e que o meu tenha ficado o mais didático possível, para ajudar a todos vocês que estão nesse mundo maravilhoso que é o *Linux*, onde todos se ajudam mutuamente.

Obrigado e espero ter contribuído com mais um artigo legal.

OBS: Como havia prometido, segue abaixo o script para inicializar o openVPN, crie o arquivo dentro do diretório /etc/init.d (isso no Conectiva 10), em outras distribuições crie dentro do diretório onde ficam os arquivos inicializáveis.

 # vi openvpn

```
#!/bin/sh
#
# openvpn       This shell script takes care of starting and stopping
#               openvpn on FreeBSD
#
# description: OpenVPN is a robust and highly flexible tunneling application that
#              uses all of the encryption, authentication, and certification features
#              of the OpenSSL library to securely tunnel IP networks over a single
#              UDP port.
#

# Contributed to the OpenVPN project by
# Douglas Keller <doug at voidstar.dyndns.org>
# 2002.05.15
# FreeBSD version by Mikhail Levin <m_levin_99 at yahoo.com>
```

```
# 2005.01.20

# The init script does the following:
#
# - Starts an openvpn process for each .conf file it finds in
#   /usr/local/etc/openvpn/config
#
# - If /usr/local/etc/openvpn/config/xxx.sh exists for a xxx.conf file then it executes
#   it before starting openvpn (useful for doing openvpn --mktun...).
#
# - In addition to start/stop you can do:
#
#   /usr/local/etc/rc.d/openvpn.sh reload - SIGHUP
#   /usr/local/etc/rc.d/openvpn.sh reopen - SIGUSR1
#   /usr/local/etc/rc.d/openvpn.sh status - SIGUSR2

# Modifications 2003.05.02
#   * Changed == to = for sh compliance (Bishop Clark).
#   * If condrestart|reload|reopen|status, check that we were
#     actually started (James Yonan).
#   * Added lock, piddir, and work variables (James Yonan).
#   * If start is attempted twice, without an intervening stop, or
#     if start is attempted when previous start was not properly
#     shut down, then kill any previously started processes, before
#     commencing new start operation (James Yonan).
#   * Do a better job of flagging errors on start, and properly
#     returning success or failure status to caller (James Yonan).

# Location of openvpn binary
openvpn="/usr/local/sbin/openvpn"

# Lockfile
lock="/var/run/lock.openvpn"

# PID directory
piddir="/var/run"

# Our working directory
work=/etc/openvpn

# Check that binary exists
if ! [ -f  $openvpn ]
then
  echo 'openvpn binary not found'
  exit 0
fi

# See how we were called.
case "$1" in

start)
  echo -n 'Starting openvpn: '

  echo -n 'if_tap '
#kldload if_tap
  echo ''

  if [ -f $lock ]
  then
   echo -n '(we were not shut down correctly) '

   for pidf in `/bin/ls $piddir/openvpn.*.pid 2>/dev/null`
   do
    if [ -s $pidf ]
    then
     kill `cat $pidf` >/dev/null 2>&1
    fi
    rm -f $pidf
   done

   rm -f $lock
   sleep 2
  fi

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

  # Start every .conf in $work and run .sh if exists

  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 $piddir/openvpn.$bn.pid
   $openvpn --daemon --writepid $piddir/openvpn.$bn.pid --config $c --cd $work
   if [ $? = 0 ]
   then
    successes=1
   else
    errors=1
   fi
  done

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

  if [ $successes = 1 ]
  then
   touch $lock
  fi
  ;;

stop)
  echo -n 'Shutting down openvpn: '

  for pidf in `/bin/ls $piddir/openvpn.*.pid 2>/dev/null`
  do
   if [ -s $pidf ]
   then
    kill `cat $pidf` >/dev/null 2>&1
   fi
   rm -f $pidf
  done
  echo -n 'success'
  rm -f $lock
  echo -n ' if_tap'
#kldunload if_tap
  echo ''
  ;;

restart)
  $0 stop
  sleep 2
  $0 start
  ;;

reload)
  if [ -f $lock ]
  then
   for pidf in `/bin/ls $piddir/openvpn.*.pid 2>/dev/null`
   do
    if [ -s $pidf ]
    then
     kill -HUP `cat $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/openvpn.*.pid 2>/dev/null`
   do
    if [ -s $pidf ]
    then
```

```
    kill -USR1 `cat $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/openvpn.*.pid 2>/dev/null`
   do
    if [ -s $pidf ]
    then
     kill -USR2 `cat $pidf` >/dev/null 2>&1
    fi
   done
   echo 'Status written to /var/log/messages'
   tail -n 3 /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
```

Salve com o comando :wq, depois execute:

 # chmod +x openvpn

Pronto, agora execute:

 # service openvpn start
ou
 # service openvpn stop

Valeu. Até a próxima!
pilao51 [a] hotmail.com
