**openvpn**

**防爆脚本安装：**

把脚本以及安装文件放到该目录下（可放在/usr/soft/下）

cd /usr/local/src

ls

chmod +x secure\_ssh.sh

mv secure\_ssh.sh /root/

cat /root/secure\_ssh.sh

#!/bin/bash

tail -100 /var/log/secure | awk '/Failed password for root/{print $(NF-3)}' | sort | uniq -c | sort -nr | awk '{print $2"="$1}' > /root/ssh\_black.list

DEFINE="5"

for i in `cat /root/ssh\_black.list`

do

IP=`echo $i | awk -F= '{print $1}'`

NUM=`echo $i | awk -F= '{print $2}'`

if [ $NUM -gt $DEFINE ]; then

grep $IP /etc/hosts.deny > /dev/null

if [ $? -gt 0 ]; then

echo "sshd:$IP:deny" >> /etc/hosts.deny

fi

fi

done

**openvpn server的安装：**

**1，依赖软件的安装：**

gcc g++ openssl openssl-devel

lzo库的编译安装

下载地址：<http://www.oberhumer.com/opensource/lzo/download/>

tar -zxvf lzo-2.10.tar.gz

cd lzo-2.10

./configure -prefix=/usr/local/lzo && make && make install

openssl安装：yum install -y openssl

**2,修改动态库的编译路径：**

vi /etc/ld.so.conf

Include /etc/ld.so.conf/\*.conf
/lib
/lib64
/usr/lib
/usr/lib64
/usr/local/lib
/usr/local/lib64

修改后，再执行:

ldconfig

**3,下载openvpn**

下载路径：http://www.openvpn.net/release/

tar -zxvf openvpn-2.1.3.tar.gz

cd openvpn-2.1.3

./configure --prefix=/usr/local/openvpn -with-lzo-headers=/usr/local/lzo/include/ -with-lzo-lib=/usr/local/lzo/lib/

echo $?

make && make install

echo $?

**4,使用openvpn源码下的工具生成证书和密钥**

mkdir /etc/openvpn

将源码目录的证书生成脚本拷贝到/etc/openvpn下面：

cp -r /usr/local/src/openvpn/easy-rsa/ /etc/openvpn/

cp /usr/local/src/openvpn-2.1.3/sample-config-files/server.conf /etc/openvpn/

cp /usr/local/src/openvpn-2.1.3/sample-config-files/client.conf /etc/openvpn/

**5,配置vars文件：**

cd /etc/openvpn/easy-rsa/2.0/

vim vars

# easy-rsa parameter settings

# NOTE: If you installed from an RPM,

# don't edit this file in place in

# /usr/share/openvpn/easy-rsa --

# instead, you should copy the whole

# easy-rsa directory to another location

# (such as /etc/openvpn) so that your

# edits will not be wiped out by a future

# OpenVPN package upgrade.

# This variable should point to

# the top level of the easy-rsa

# tree.

export EASY\_RSA="`pwd`"

#

# This variable should point to

# the requested executables

#

export OPENSSL="openssl"

export PKCS11TOOL="pkcs11-tool"

export GREP="grep"

# This variable should point to

# the openssl.cnf file included

# with easy-rsa.

export KEY\_CONFIG=`$EASY\_RSA/whichopensslcnf $EASY\_RSA`

# Edit this variable to point to

# your soon-to-be-created key

# directory.

#

# WARNING: clean-all will do

# a rm -rf on this directory

# so make sure you define

# it correctly!

export KEY\_DIR="$EASY\_RSA/keys"

# Issue rm -rf warning

echo NOTE: If you run ./clean-all, I will be doing a rm -rf on $KEY\_DIR

# PKCS11 fixes

export PKCS11\_MODULE\_PATH="dummy"

export PKCS11\_PIN="dummy"

# Increase this to 2048 if you

# are paranoid. This will slow

# down TLS negotiation performance

# as well as the one-time DH parms

# generation process.

export KEY\_SIZE=1024

# In how many days should the root CA key expire?

export CA\_EXPIRE=3650

# In how many days should certificates expire?

export KEY\_EXPIRE=3650

# These are the default values for fields

# which will be placed in the certificate.

# Don't leave any of these fields blank.

export KEY\_COUNTRY="CN"

export KEY\_PROVINCE="SC"

export KEY\_CITY="chengdu"

export KEY\_ORG="Healthy"

export KEY\_EMAIL="me@myhost.mydomain"

并执行

source ./vars

**6,生成证书**

清空原来的认证文件，进行重新生成：

./clean-all

生成ca文件：

./build-ca

生成Diffie Hellman文件：

./build-dh

生成服务端证书文件：

./build-key-server server

生成客户端test1的证书文件:./build-key test1

生成ta.key防止遭到DDOS攻击：openvpn --genkey --secret ta.key

上面生成的文件，都在keys目录下面

**7,下面是修改配置server、cilent执行文件：**

**vim /etc/openvpn/server.conf**

#################################################

# Sample OpenVPN 2.0 config file for #

# multi-client server. #

# #

# This file is for the server side #

# of a many-clients <-> one-server #

# OpenVPN configuration. #

# #

# OpenVPN also supports #

# single-machine <-> single-machine #

# configurations (See the Examples page #

# on the web site for more info). #

# #

# This config should work on Windows #

# or Linux/BSD systems. Remember on #

# Windows to quote pathnames and use #

# double backslashes, e.g.: #

# "C:\\Program Files\\OpenVPN\\config\\foo.key" #

# #

# Comments are preceded with '#' or ';' #

#################################################

# Which local IP address should OpenVPN

# listen on? (optional)

;local a.b.c.d

# Which TCP/UDP port should OpenVPN listen on?

# If you want to run multiple OpenVPN instances

# on the same machine, use a different port

# number for each one. You will need to

# open up this port on your firewall.

port 1433

# TCP or UDP server?

proto tcp

;proto udp

# "dev tun" will create a routed IP tunnel,

# "dev tap" will create an ethernet tunnel.

# Use "dev tap0" if you are ethernet bridging

# and have precreated a tap0 virtual interface

# and bridged it with your ethernet interface.

# If you want to control access policies

# over the VPN, you must create firewall

# rules for the the TUN/TAP interface.

# On non-Windows systems, you can give

# an explicit unit number, such as tun0.

# On Windows, use "dev-node" for this.

# On most systems, the VPN will not function

# unless you partially or fully disable

# the firewall for the TUN/TAP interface.

dev tap

;dev tun

# Windows needs the TAP-Win32 adapter name

# from the Network Connections panel if you

# have more than one. On XP SP2 or higher,

# you may need to selectively disable the

# Windows firewall for the TAP adapter.

# Non-Windows systems usually don't need this.

;dev-node MyTap

# SSL/TLS root certificate (ca), certificate

# (cert), and private key (key). Each client

# and the server must have their own cert and

# key file. The server and all clients will

# use the same ca file.

#

# See the "easy-rsa" directory for a series

# of scripts for generating RSA certificates

# and private keys. Remember to use

# a unique Common Name for the server

# and each of the client certificates.

#

# Any X509 key management system can be used.

# OpenVPN can also use a PKCS #12 formatted key file

# (see "pkcs12" directive in man page).

ca /etc/openvpn/keys/ca.crt

cert /etc/openvpn/keys/server.crt

key /etc/openvpn/keys/server.key # This file should be kept secret

# Diffie hellman parameters.

# Generate your own with:

# openssl dhparam -out dh1024.pem 1024

# Substitute 2048 for 1024 if you are using

# 2048 bit keys.

dh /etc/openvpn/keys/dh1024.pem

# Configure server mode and supply a VPN subnet

# for OpenVPN to draw client addresses from.

# The server will take 10.8.0.1 for itself,

# the rest will be made available to clients.

# Each client will be able to reach the server

# on 10.8.0.1. Comment this line out if you are

# ethernet bridging. See the man page for more info.

server 10.8.0.0 255.255.255.0

# Maintain a record of client <-> virtual IP address

# associations in this file. If OpenVPN goes down or

# is restarted, reconnecting clients can be assigned

# the same virtual IP address from the pool that was

# previously assigned.

ifconfig-pool-persist ipp.txt

# Configure server mode for ethernet bridging.

# You must first use your OS's bridging capability

# to bridge the TAP interface with the ethernet

# NIC interface. Then you must manually set the

# IP/netmask on the bridge interface, here we

# assume 10.8.0.4/255.255.255.0. Finally we

# must set aside an IP range in this subnet

# (start=10.8.0.50 end=10.8.0.100) to allocate

# to connecting clients. Leave this line commented

# out unless you are ethernet bridging.

;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# Configure server mode for ethernet bridging

# using a DHCP-proxy, where clients talk

# to the OpenVPN server-side DHCP server

# to receive their IP address allocation

# and DNS server addresses. You must first use

# your OS's bridging capability to bridge the TAP

# interface with the ethernet NIC interface.

# Note: this mode only works on clients (such as

# Windows), where the client-side TAP adapter is

# bound to a DHCP client.

;server-bridge

# Push routes to the client to allow it

# to reach other private subnets behind

# the server. Remember that these

# private subnets will also need

# to know to route the OpenVPN client

# address pool (10.8.0.0/255.255.255.0)

# back to the OpenVPN server.

push "route 172.27.16.0 255.255.240.0"

;push "route 192.168.20.0 255.255.255.0"

# To assign specific IP addresses to specific

# clients or if a connecting client has a private

# subnet behind it that should also have VPN access,

# use the subdirectory "ccd" for client-specific

# configuration files (see man page for more info).

# EXAMPLE: Suppose the client

# having the certificate common name "Thelonious"

# also has a small subnet behind his connecting

# machine, such as 192.168.40.128/255.255.255.248.

# First, uncomment out these lines:

;client-config-dir ccd

;route 192.168.40.128 255.255.255.248

# Then create a file ccd/Thelonious with this line:

# iroute 192.168.40.128 255.255.255.248

# This will allow Thelonious' private subnet to

# access the VPN. This example will only work

# if you are routing, not bridging, i.e. you are

# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give

# Thelonious a fixed VPN IP address of 10.9.0.1.

# First uncomment out these lines:

;client-config-dir ccd

;route 10.9.0.0 255.255.255.252

# Then add this line to ccd/Thelonious:

# ifconfig-push 10.9.0.1 10.9.0.2

# Suppose that you want to enable different

# firewall access policies for different groups

# of clients. There are two methods:

# (1) Run multiple OpenVPN daemons, one for each

# group, and firewall the TUN/TAP interface

# for each group/daemon appropriately.

# (2) (Advanced) Create a script to dynamically

# modify the firewall in response to access

# from different clients. See man

# page for more info on learn-address script.

;learn-address ./script

# If enabled, this directive will configure

# all clients to redirect their default

# network gateway through the VPN, causing

# all IP traffic such as web browsing and

# and DNS lookups to go through the VPN

# (The OpenVPN server machine may need to NAT

# or bridge the TUN/TAP interface to the internet

# in order for this to work properly).

;push "redirect-gateway def1 bypass-dhcp"

# Certain Windows-specific network settings

# can be pushed to clients, such as DNS

# or WINS server addresses. CAVEAT:

# http://openvpn.net/faq.html#dhcpcaveats

# The addresses below refer to the public

# DNS servers provided by opendns.com.

;push "dhcp-option DNS 208.67.222.222"

;push "dhcp-option DNS 208.67.220.220"

# Uncomment this directive to allow different

# clients to be able to "see" each other.

# By default, clients will only see the server.

# To force clients to only see the server, you

# will also need to appropriately firewall the

# server's TUN/TAP interface.

client-to-client

# Uncomment this directive if multiple clients

# might connect with the same certificate/key

# files or common names. This is recommended

# only for testing purposes. For production use,

# each client should have its own certificate/key

# pair.

#

# IF YOU HAVE NOT GENERATED INDIVIDUAL

# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,

# EACH HAVING ITS OWN UNIQUE "COMMON NAME",

# UNCOMMENT THIS LINE OUT.

duplicate-cn

# The keepalive directive causes ping-like

# messages to be sent back and forth over

# the link so that each side knows when

# the other side has gone down.

# Ping every 10 seconds, assume that remote

# peer is down if no ping received during

# a 120 second time period.

keepalive 10 120

# For extra security beyond that provided

# by SSL/TLS, create an "HMAC firewall"

# to help block DoS attacks and UDP port flooding.

#

# Generate with:

# openvpn --genkey --secret ta.key

#

# The server and each client must have

# a copy of this key.

# The second parameter should be '0'

# on the server and '1' on the clients.

tls-auth /etc/openvpn/keys/ta.key 0 # This file is secret

# Select a cryptographic cipher.

# This config item must be copied to

# the client config file as well.

;cipher BF-CBC # Blowfish (default)

cipher AES-256-CBC # AES

;cipher DES-EDE3-CBC # Triple-DES

# Enable compression on the VPN link.

# If you enable it here, you must also

# enable it in the client config file.

comp-lzo

# The maximum number of concurrently connected

# clients we want to allow.

;max-clients 100

# It's a good idea to reduce the OpenVPN

# daemon's privileges after initialization.

#

# You can uncomment this out on

# non-Windows systems.

;user nobody

;group nobody

# The persist options will try to avoid

# accessing certain resources on restart

# that may no longer be accessible because

# of the privilege downgrade.

persist-key

persist-tun

# Output a short status file showing

# current connections, truncated

# and rewritten every minute.

status openvpn-status.log

# By default, log messages will go to the syslog (or

# on Windows, if running as a service, they will go to

# the "\Program Files\OpenVPN\log" directory).

# Use log or log-append to override this default.

# "log" will truncate the log file on OpenVPN startup,

# while "log-append" will append to it. Use one

# or the other (but not both).

;log openvpn.log

;log-append openvpn.log

# Set the appropriate level of log

# file verbosity.

#

# 0 is silent, except for fatal errors

# 4 is reasonable for general usage

# 5 and 6 can help to debug connection problems

# 9 is extremely verbose

verb 3

# Silence repeating messages. At most 20

# sequential messages of the same message

# category will be output to the log.

;mute 20

**vim /etc/openvpn/client.conf**

##############################################

# Sample client-side OpenVPN 2.0 config file #

# for connecting to multi-client server. #

# #

# This configuration can be used by multiple #

# clients, however each client should have #

# its own cert and key files. #

# #

# On Windows, you might want to rename this #

# file so it has a .ovpn extension #

##############################################

# Specify that we are a client and that we

# will be pulling certain config file directives

# from the server.

client

# Use the same setting as you are using on

# the server.

# On most systems, the VPN will not function

# unless you partially or fully disable

# the firewall for the TUN/TAP interface.

dev tap

;dev tun

# Windows needs the TAP-Win32 adapter name

# from the Network Connections panel

# if you have more than one. On XP SP2,

# you may need to disable the firewall

# for the TAP adapter.

;dev-node MyTap

# Are we connecting to a TCP or

# UDP server? Use the same setting as

# on the server.

proto tcp

;proto udp

# The hostname/IP and port of the server.

# You can have multiple remote entries

# to load balance between the servers.

remote 118.24.18.114 1433

;remote my-server-2 1194

# Choose a random host from the remote

# list for load-balancing. Otherwise

# try hosts in the order specified.

;remote-random

# Keep trying indefinitely to resolve the

# host name of the OpenVPN server. Very useful

# on machines which are not permanently connected

# to the internet such as laptops.

resolv-retry infinite

# Most clients don't need to bind to

# a specific local port number.

nobind

# Downgrade privileges after initialization (non-Windows only)

;user nobody

;group nobody

# Try to preserve some state across restarts.

persist-key

persist-tun

# If you are connecting through an

# HTTP proxy to reach the actual OpenVPN

# server, put the proxy server/IP and

# port number here. See the man page

# if your proxy server requires

# authentication.

;http-proxy-retry # retry on connection failures

;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot

# of duplicate packets. Set this flag

# to silence duplicate packet warnings.

;mute-replay-warnings

# 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

# Verify server certificate by checking that the

# certicate has the correct key usage set.

# This is an important precaution to protect against

# a potential attack discussed here:

# http://openvpn.net/howto.html#mitm

#

# To use this feature, you will need to generate

# your server certificates with the keyUsage set to

# digitalSignature, keyEncipherment

# and the extendedKeyUsage to

# serverAuth

# EasyRSA can do this for you.

remote-cert-tls server

# If a tls-auth key is used on the server

# then every client must also have the key.

tls-auth ta.key 1

# Select a cryptographic cipher.

# If the cipher option is used on the server

# then you must also specify it here.

# Note that v2.4 client/server will automatically

# negotiate AES-256-GCM in TLS mode.

# See also the ncp-cipher option in the manpage

cipher AES-256-CBC

# Enable compression on the VPN link.

# Don't enable this unless it is also

# enabled in the server config file.

comp-lzo

# Set log file verbosity.

verb 3

dhcp-option DNS 119.29.29.29

# Silence repeating messages

;mute 20

**启动服务：**

openvpn --config /etc/openvpn/server.conf

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

杀服务：

ps aux |grep openvpn

kill -9 进程号

**修改路由规则：**

vim /etc/sysctl.conf

net.ipv4.ip\_forward = 1

sysctl -p

**如需重新生成**

cd /usr/local/src/openvpn-2.1.3/easy-rsa/2.0/

source vars

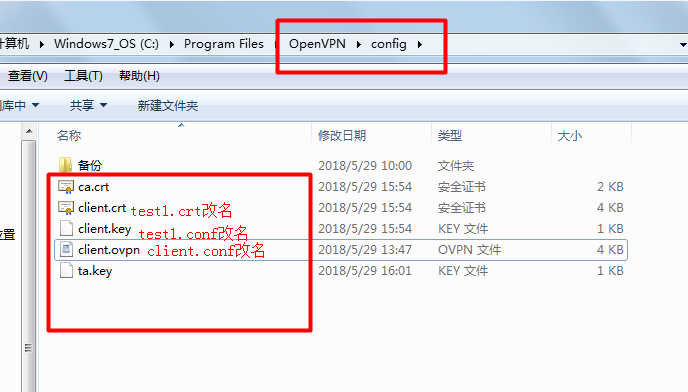
./build-key zc

cp zc.\* /etc/openvpn/keys

**客户端安装**

安装客户端，执行exe程序，默认下一步，将文件从服务器上拷贝下来 放入config内改名即可

在客户机上，我们需要保证客户机的文件有以下的文件：



*port 1194*

*proto tcp*

*dev tun*

*ca /etc/openvpn/easy-rsa/2.0/keys/ca.crt*

*cert /etc/openvpn/easy-rsa/2.0/keys/server.crt*

*key /etc/openvpn/easy-rsa/2.0/keys/server.key # This file should be kept secret*

*dh /etc/openvpn/easy-rsa/2.0/keys/dh1024.pem ##加密文件路径*

*server 10.8.0.0 255.255.255.0 ##指定VPN子网段，10.8.0.1为服务器IP，其余分给客户端*

*ifconfig-pool-persist ipp.txt ##记录客户端所获得的IP，在里面可以给客户端指定IP*

*client-to-client ##允许客户机之间互相连通*

*duplicate-cn ##允许多个客户端使用一个证书和密钥，当使用相同证书和密码的客户端同时连接时，可能会造成VPN掉线*

*keepalive 10 120 ##每隔120秒测试连接可用性，每次ping10秒*

*tls-auth /etc/openvpn/easy-rsa/2.0/keys/ta.key 0 # This file is secret*

*cipher AES-256-CBC ##客户端必须使用相同配置*

*comp-lzo ##开启VPN连接压缩，客户端必须做相同设置*

*persist-key*

*persist-tun*

*status openvpn-status.log*

*verb 3*

*client.conf文件配置：*

*client*

*dev tun*

*proto tcp*

*remote 39.108.78.127 1194*

*resolv-retry infinite*

*nobind*

*persist-key*

*persist-tun*

*ca /etc/openvpn/client/ca.crt*

*cert /etc/openvpn/client/client.crt*

*key /etc/openvpn/client/client.key*

*remote-cert-tls server*

*tls-auth /etc/openvpn/client/ta.key 1*

*cipher AES-256-CBC*

*comp-lzo*

*verb 3*

*dhcp-option DNS 119.29.29.29*

启动方法：

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

在启动的时候出现错误，该怎么排查？怎么查看是否执行成功？

若检测未开启服务，则说明服务端配置参数有误，去掉–daemon运行参数

若运行成功，执行ifconfig命令会多出一个tun0,即VPN虚拟网卡

在我们遇到错误的时候，我们需要查看我们的错误日志，请将--dadmon这个参数去掉