### vpn服务器操作手册

**前言：**

1.蓝色字体表示命令行命令，正式执行时不要复制前面的#号，#号只是提示应该使用root权限操作

2.绿色字体表示注释，有时注释太多就不用绿色表示了

3.注意：本文档的所有操作请先在测环境进行实践，请不要直接在真实的服务器中操作！

**版权声明**：

本文档以开源的形式发布，所有条款如下：

1. 无担保：作者不保证文档内容的准确无误，亦不承担由于使用此文档所导致的任何后果

2. 自由使用：任何人可以出于任何目的而自由地 阅读/链接/打印/转载/引用/再创作 此文档，无需任何附加条件

若您 阅读/链接/打印/转载/引用/再创作 本文档，则说明接受以上2个条款。

作者：李茂福

时间：2023-06-13

**第0章、VPN简介**

**VPN解决的问题：**

①不同的用户可使用重叠（相同）的ip地址段，在转发时互不干扰，路由表隔离开来

②用户的ip地址不暴露在外网上，是以隧道的方式传递的（将用户流量封装在外网流量之上）

**VPN隧道分类：**

①pptp/l2tp/vxlan这些是将用户的二层流量（mac帧）封装在外网的udp报文之上，也称为二层隧道协议

②ipsec这种是将用户的三层流量（ip报文）封装在外网的ip报文之上，也称为三层隧道协议

③mpls vpn这种是在用户的三层流量之前插入一个mpls Label，在外网上只根据mpls label进行转发，也称为第二层之上和第三层之下的隧道技术

④OpenVPN既支持将用户的二层流量封装在传输层之上，也可将用户的三层流量封装在传输层之上（传输层可用tcp也可用udp）

**外网**是指不属于用户内部网络的网络，是用来承载用户内部网络流量并打通用户不同站点之间的通信（用于承载用户要走隧道的流量）

**第一章、L2TP VPN**

在

**第二章、ipsec（strongswan）**

在epel源里有strongswan的安装包

# cat /etc/yum.repos.d/epel.repo

[epel]

name=Extra Packages for Enterprise Linux 7 - $basearch

#baseurl=http://download.fedoraproject.org/pub/epel/7/$basearch

metalink=https://mirrors.fedoraproject.org/metalink?repo=epel-7&arch=$basearch

failovermethod=priority

enabled=1

gpgcheck=1

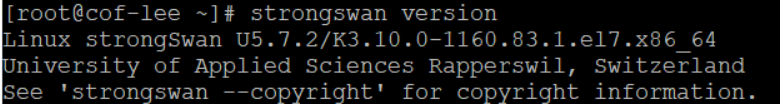
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7

# yum install epel-release #安装epel的yum源

# yum install strongswan #安装strongswan

#依赖trousers包

# strongswan version #查看安装的版本



# vi /etc/strongswan/ipsec.conf #主配置文件

|  |  |
| --- | --- |
| config setup |  |
| uniqueids=never | 关闭ID唯一性，即允许多个客户端使用同一个证书，多设备同时在线 |
| conn %default | 默认配置项，其他conn配置项都会调用此默认项 |
| authby=psk | 默认使用预共享密钥认证方式 |
| type=tunnel |  |
| conn to-west |  |
| left=x.x.x.x | 本vpn服务器的公网IP地址，如果与对端vpn服务器之间不经过nat设备，则直接写本vpn服务器的内网ip |
| leftid=x.x.x.x | 本地网关设备的标识，本vpn服务器的内网ip |
| leftsubnet=192.168.0.0/24 | 本vpn服务器这一侧的网段，本地端网络，0.0.0.0/0为通配，表示所有IP网段 |
| right=y.y.y.y | 对端vpn服务器的公网IP地址，不经过公网则直接写内网ip |
| rightid=y.y.y.y | 对端VPN网关的标识，对端vpn服务器的内网ip |
| rightsubnet=192.168.100.0/24 | 对端vpn服务器那一侧的网段 |
| dpdaction=clear | dpd表示Dead Peer Detection，对端失效检测，clear表示对端失效时关闭连接 |
| keyexchange=ikev1 | IPsec连接使用的IKE版本 |
| auto=route |  |
| ike=aes128-sha256-modp2048 | IKE协议的加密算法-认证算法-DH分组 |
| ikelifetime=86400s | IKE协议的SA生命周期 |
| esp=aes128-sha256-modp2048 | IPsec协议的加密算法-认证算法-DH分组 |
| lifetime=86400s | IPsec协议的SA生命周期 |
| type=tunnel |  |
| fragmentation=yes | 允许分片 |

**★ike及esp加密认证算法还可为：（多个值之间用逗号隔开）**

aes128-sha1-modp1024,aes128-sha1-modp2048,aes128-sha256-modp1024,aes128-sha256-modp2048,aes256-sha256-modp1024,aes256-sha256-modp2048,aes256-sha256-modp4096,aes128gcm12-aes128gcm16-aes256gcm12-aes256gcm16-modp2048-modp4096-modp1024

**★使用ssl证书认证：**

leftauth=pubkey #本地端使用公钥验证

leftcert= x.x.x.x.cert.pem #指定本地端证书

rightauth=pubkey #远程端使用公钥验证

rightcert= y.y.y.y.cert.pem #指定远程端证书

# vi /etc/strongswan/ipsec.secrets

x.x.x.x y.y.y.y : PSK passxx123

# x.x.x.x为leftid，y.y.y.y为rightid

# passxx123为IPsec连接的预共享密钥，2端的vpn服务器预共享密钥要求一致

# vi /etc/strongswan/strongswan.d/charon.conf

max\_ikev1\_exchanges = 50 #取消前面的#注释，并修改为允许建立的最大连接数，默认为最大只允许建立3条连接

**★开启系统ipv4路由转发**

# vi /etc/sysctl.conf

net.ipv4.ip\_forward=1

# echo 1 > /proc/sys/net/ipv4/ip\_forward

**★防火墙放通端口及协议报文入站**

firewall-cmd --add-port=500/udp

firewall-cmd --add-port=4500/udp

firewall-cmd --add-rich-rule='rule protocol value="esp" accept'

firewall-cmd --add-rich-rule='rule protocol value="ah" accept'

firewall-cmd --runtime-to-permanent

★启动服务

# systemctl enable strongswan

# systemctl start strongswan

# strongswan status #查看服务状态

# strongswan listcacerts #查看加载的CA证书

**第三章、OpenVPN**

下载地址： https://build.openvpn.net/downloads/releases/

windows建议版本：

openvpn-install-2.4.9-I601-Win10.exe

tap-windows-9.24.2-I601-Win10.exe

windows安装目录： "C:\Program Files\OpenVPN\"

demo配置目录： "C:\Program Files\OpenVPN\sample-config\"

主配置目录： "C:\Program Files\OpenVPN\config\"

★centos7安装

# yum install epel-release

# yum install openvpn

启动命令：

# /usr/sbin/openvpn --daemon --config /etc/openvpn/server/serv.ovpn

**★准备材料**

tls-auth.key

dh2048.pem

ovpn.ca.crt

ovpn.server.crt

ovpn.server.key

ovpn.client1.crt

ovpn.client1.key

ovpn.crl.pem

**#生成tls-auth.key及dh2048.pem命令：**

# /usr/sbin/openvpn --genkey --secret tls-auth.key

# openssl dhparam -out dh2048.pem 2048

**★创建CA证书**

#创建证书前一定要确认系统时间是否正确，以免创建出的证书因为时间不对而未生效

# CA\_DOMAIN\_NAME=ovpn.ca

# openssl genrsa -out ${CA\_DOMAIN\_NAME}.key 2048

# openssl req -new -sha256 -key ${CA\_DOMAIN\_NAME}.key \

-subj "/C=CN/ST=GD/L=SZ/O=coflee/OU=it/CN=${CA\_DOMAIN\_NAME}" \

-out ${CA\_DOMAIN\_NAME}.csr

# cat > ${CA\_DOMAIN\_NAME}.ssl.conf <<EOF

authorityKeyIdentifier=keyid,issuer

basicConstraints=CA:TRUE

keyUsage=digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment,keyAgreement

extendedKeyUsage=serverAuth,clientAuth

subjectAltName=@alt\_names

[alt\_names]

DNS.1=${CA\_DOMAIN\_NAME}

EOF

# openssl x509 -req -sha256 -days 3650 -signkey ${CA\_DOMAIN\_NAME}.key \

-extfile ${CA\_DOMAIN\_NAME}.ssl.conf -in ${CA\_DOMAIN\_NAME}.csr \

-out ${CA\_DOMAIN\_NAME}.crt

# openssl x509 -text -noout -in ${CA\_DOMAIN\_NAME}.crt #查看证书内容

**★创建server/client证书**

# DOMAIN\_NAME=ovpn.server

# openssl genrsa -out ${DOMAIN\_NAME}.key 2048

# openssl req -new -sha256 -key ${DOMAIN\_NAME}.key \

-subj "/C=CN/ST=GD/L=SZ/O=coflee/OU=it/CN=${DOMAIN\_NAME}" \

-out ${DOMAIN\_NAME}.csr

# cat > ${DOMAIN\_NAME}.ssl.conf <<EOF

authorityKeyIdentifier=keyid,issuer

basicConstraints=CA:FALSE

keyUsage=digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment,keyAgreement

extendedKeyUsage=serverAuth,clientAuth

subjectAltName=@alt\_names

[alt\_names]

DNS.1=${DOMAIN\_NAME}

EOF

#用CA证书去给其他server签名证书

# openssl x509 -req -sha256 -days 3649 -CA ${CA\_DOMAIN\_NAME}.crt \

-CAkey ${CA\_DOMAIN\_NAME}.key -CAserial ${CA\_DOMAIN\_NAME}.srl \

-CAcreateserial -extfile ${DOMAIN\_NAME}.ssl.conf \

-in ${DOMAIN\_NAME}.csr -out ${DOMAIN\_NAME}.crt

**★TAP和TUN设备**

TAP为二层设备，底层为mac帧，有二层广播包，不可用于Andriod/ios手机端，tap设备可放入bridge网桥中

TUN为三层设备，底层为ip报文，仅ipv4/ipv6报文

MTU设置：

TAP 1426

TUN 1444

**★server配置文件**

# vi /etc/openvpn/server/serv.ovpn

|  |  |
| --- | --- |
| local x.x.x.x | 指定监听的ip，默认监听所有ip |
| port 7800 |  |
| proto tcp | 默认为1194/udp |
| dev tun | 或者dev tap，要与客户端一致 |
| ca /etc/openvpn/server/ovpn.ca.crt |  |
| cert /etc/openvpn/server/ovpn.server.crt |  |
| key /etc/openvpn/server/ovpn.server.key |  |
| dh /etc/openvpn/server/dh2048.pem |  |
| tls-auth /etc/openvpn/server/tls-auth.key |  |
| topology subnet |  |
| server 10.8.8.0 255.255.255.0 | 给客户端使用的私有网段 |
| ifconfig-pool-persist /etc/openvpn/server/ipp.txt | 记录客户端分到的ip |
| duplicate-cn | 允许多个客户端使用相同的ssl证书 |
| client-to-client | 允许各个客户端之间互相通信 |
| keepalive 30 60 | 每30秒检测客户端是否在线，超60秒无回复则认为下线了 |
| comp-lzo | 全局一致，使用压缩，服务端与客户端都开或都关 |
| user nobody | 仅linux端配置，降级权限 |
| group nobody | 仅linux端配置，降级权限 |
| persist-key |  |
| persist-tun |  |
| status /etc/openvpn/server/ovpn-status.log | 状态记录 |
| log-append /etc/openvpn/server/ovpn.log | 服务日志 |
| verb 4 | 日志级别，0~9，0为不记录日志 |
| # crl-verify /etc/openvpn/server/ovpn.crl.pem | crl证书吊销列表文件 |
| max-clients 100 | 最大客户端数量 |
| push "route 10.9.10.0 255.255.255.0" | 推送路由给客户端，可有多条，一行一条 |
| push "dhcp-option DNS 8.8.8.8" | 推送DNS |
| push "redirect-gateway def1 bypass-dhcp" | 推送默认路由，让所有客户端的默认路由都指向ovpn |

**#启动命令：**

# /usr/sbin/openvpn --daemon --config /etc/openvpn/server/serv.ovpn

**★client配置文件**

# vi /etc/openvpn/client/client1.ovpn

|  |  |
| --- | --- |
| client |  |
| dev tun |  |
| proto tcp |  |
| remote 服务端ip 7800 |  |
| resolv-retry infinite |  |
| dev-node TAP名称 | windows上要安装TAP网卡，若有多个TAP设备，则需要指明要使用的tap |
| nobind | 不使用固定tcp端口 |
| persist-key |  |
| persist-tun |  |
| user nobody | 仅linux端配置，降级权限 |
| group nobody | 仅linux端配置，降级权限 |
| ca /etc/openvpn/client/ovpn.ca.crt |  |
| cert /etc/openvpn/client/ovpn.client1.crt |  |
| key /etc/openvpn/client/ovpn.client1.key |  |
| tls-auth /etc/openvpn/client/tls-auth.key 1 |  |
| # remote-cert-tls server | 检查服务端证书的用法（digital Signature, keyEncipherment） |
| ciper AES-256-GCM | 默认的，可指定为AES-256-CBC等 |
| comp-lzo | 需与服务端保持一致 |
| verb 4 |  |
| auth-nocache | 不保存密码在cache中 |
| # http-proxy x.x.x.x 10809 | 设置让ovpn流量走HTTP代理，此代理不用验证 |
| route-nopull | 不从server端获取路由信息 |
| route x.x.x.x 255.255.255.0 | 自己设置客户端的路由 |
|  |  |
|  |  |

cmd> "C:\\Program Files\\OpenVPN\\bin\\openvpn.exe" --daemon --config "C:\\Program Files\\OpenVPN\\config\\client1.ovpn"

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

#客户端的配置文件及证书等可合并为一个单一文件：

client

dev tun

proto tcp

......

<key>

xxx

</key>

<cert>

xxx

</cert>

<ca>

xxx

</ca>

key-direction 1 #tls-auth用法，1

<tls-auth>

xxx

</tls-auth>

★给特定client分配固定ip

# vi /etc/openvpn/server/serv.ovpn #新增以下配置

client-config-dir /etc/openvpn/server/ccd #一个子目录

#在/etc/openvpn/server/ccd目录下创建一个文件，文件名为客户端ssl证书里的CN名称，如： client1.cof-lee.com

# vi /etc/openvpn/server/ccd/client1.cof-lee.com #内容就以下一行

ifconfig-push 10.8.8.44 255.255.255.0

★客户端使用账号密码验证

默认是要求使用ssl证书验证，可以设置为仅使用账号密码验证

# wget http://openvpn.se/files/other/checkpsw.sh -O /etc/openvpn/server/checkpsw.sh

# chmod +x /etc/openvpn/server/checkpsw.sh

# vi /etc/openvpn/server/serv.ovpn

auth-user-pass verify /etc/openvpn/server/checkpsw.sh via-env

script-security 3 # openvpn --help | grep -A s script-security

client-cert-not-required #必须指明客户端不使用ssl证书，否则除了要输入账号密码，仍然要ssl证书

username-as-common-name

# vi /etc/openvpn/server/checkpsw.sh #修改以下2个参数

PASSFILE="/etc/openvpn/server/client-psw-file"

LOG\_FILE="/etc/openvpn/server/client-psw.log"

......

# touch /etc/openvpn/server/client-psw.log #提前创建日志文件

# vi /etc/openvpn/server/client-psw-file

userName1 passwd1

userName2 passwd2

# chown nobody:nobody /etc/openvpn/server/client-psw-file

## 客户端配置

# vi clientxx.ovpn

ca /xxx/ca.crt #校验ca证书

auth-user-pass #使用账号密码验证

**★checkpsw.sh文件内容：**

#!/bin/sh

PASSFILE="/etc/openvpn/psw-file"

LOG\_FILE="/var/log/openvpn-password.log"

TIME\_STAMP=`date "+%Y-%m-%d %T"`

if [ ! -r "${PASSFILE}" ]; then

echo "${TIME\_STAMP}: Could not open password file \"${PASSFILE}\" for reading." >> ${LOG\_FILE}

exit 1

fi

CORRECT\_PASSWORD=`awk '!/^;/&&!/^#/&&$1=="'${username}'"{print $2;exit}' ${PASSFILE}`

if [ "${CORRECT\_PASSWORD}" = "" ]; then

echo "${TIME\_STAMP}: User does not exist: username=\"${username}\", password=\"${password}\"." >> ${LOG\_FILE}

exit 1

fi

if [ "${password}" = "${CORRECT\_PASSWORD}" ]; then

echo "${TIME\_STAMP}: Successful authentication: username=\"${username}\"." >> ${LOG\_FILE}

exit 0

fi

echo "${TIME\_STAMP}: Incorrect password: username=\"${username}\", password=\"${password}\"." >> ${LOG\_FILE}

exit 1

#