MOBILE VPN con OpenVPN - Zeroshell

e la partecipazione straordinaria
di
Android e Ipad

Il sistema operativo multifunzionale
creato da Fulvio.Ricciardi@zeroshell.net
www.zeroshell.net

OpenVpn in più di 2 minuti per configurare Zeroshell per Android ed Ipad
(Autore: fnc.gulino@libero.it)

Ho conosciuto, scoperto e apprezzato Zeroshell molti anni fa, ho fatto buon uso delle sue qualità ed esso mi ha ricambiato sempre con la sua solidità, semplicità e tranquilla dedizione.
Fino a quando mi sono imbattuto nella "patata bollente" della connessione in openvpn dei terminali mobili, ed in particolare di Smartphone Android ed Ipad (mini e maxi), che mi ha tolto il sorriso e la convinzione che nulla fosse impossibile per il magnifico Zeroshell.
Da premettere che utilizzo openvpn in implementazioni non zeroshelliane da tempo immemore e la sua configurazione in tutte le "salse" ha sempre risposto alle mie esigenze.
Visti i primi fallimenti ho girovagato nel forum ed in rete per scoprire le ragioni dei miei errori e della mia non sempre manifesta ignoranza, ho scoperto i link che mi hanno guidato alla soluzione ed al ritorno dell'allegria nei miei poveri occhi stanchi.
Certo di fare cosa simpatica e gradita mi permetterò di divulgare questa mia semplice esperienza.

"Grazie Fulvio, e grazie a tutti quelli che in questi anni sono stati artefici della creazione e sviluppo di un sistema operativo di controllo e gestione rete veramente eccezionale."

Partiamo!

In rete e nel forum si può far riferimento in particolare a questa documentazione per stabilire una connessione openvpn con android o con ipad:

1) http://www.zeroshell.org/forum/viewtopic.php?p=4805#4805
2) http://www.zeroshell.org/forum/viewtopic.php?t=3910&sid=d45e8047eedb9355c02b4ebfa025b27e,

in effetti una volta stabilito che i terminali in questione possono solo utilizzare i tunnel con device di tipo ***"tun"***, basta fare qualche tentativo per modificare e trovare le configurazioni corrette da dare in pasto ai client Vpn (Openvpn Connect Android e Openvn Connect Ipad) ed al server openvpn di Zeroshell.
Si suggerisce al 2) di sostituire negli script di avvio "`/root/kerbynet.cgi/scripts`" le occorrenze "`grep -w tap *`" con il corrispettivo "`tun`", infine per rendere stabile la configurazione al riavvio si consiglia l'uso dei post boot script.

3) http://www.zeroshell.net/forum/viewtopic.php?p=14404&sid=4e28ef84d3186cbe411bb2f03510194b.

Fin qui tutto bene! Il problema sorge invece se si vuole mantenere intatta la funzionalita ***"tap"*** utilissima e comodissima in contesto "Host-to-LAN" con Zeroshell, oppure mantenere intatta la capacità di utilizzare l'interfaccia web di gestione per controllo e manutenzione delle vpn create.

La soluzione testata e che di seguito espongo, penso sia utile, per estendere le già notevoli "capacità" openvpnenniane del caro Zeroshell.

Iniziamo con modificare gli script in questione, evidenziate in giallo le semplici aggiunte, che in sostanza credo non snaturino lo spirito di partenza del progetto:

"createvpn";

```
#!/bin/sh
. /etc/kerbynet.conf
. $SCRIPTS/net.inc
NAME="$1"
DESCRIPTION="$2"
REMOTEIP="$3"
PORT="$4"
PROTO="$5"
TLSROLE="$6"
REMOTECN="$7"
COMPRESSION="$8"
CRYPTO="$9"
```

```
PARAMETERS="${10}"
AUTHENTICATION="${11}"
PSK="${12}"
GATEWAY="${13}"
LOCALIP="${14}"
[ -z "$NAME" ] && exit 1
CONFIG="$REGISTER/system/net/interfaces/"
VPNDIR="$CONFIG/$NAME"
if [ "$NAME" == VPN00  ] ; then
        DEV_TYPE="tun"
else
        DEV_TYPE="tap"
fi
if openvpn --dev-type $DEV_TYPE --dev $NAME --mktun 1>&2 ; then
   mkdir -p $VPNDIR/TUNNEL
   echo "$DESCRIPTION" > $VPNDIR/Description
   echo "$REMOTEIP" > $VPNDIR/TUNNEL/RemoteIP
   echo "$LOCALIP" > $VPNDIR/TUNNEL/LocalIP
   echo "$PORT" > $VPNDIR/TUNNEL/Port
   echo "$PROTO" > $VPNDIR/TUNNEL/Proto
   echo "$TLSROLE" > $VPNDIR/TUNNEL/TLSRole
   echo "$REMOTECN" > $VPNDIR/TUNNEL/RemoteCN
   echo "$COMPRESSION" > $VPNDIR/TUNNEL/Compression
   echo "$CRYPTO" > $VPNDIR/TUNNEL/Crypto
   echo "$AUTHENTICATION" > $VPNDIR/TUNNEL/Authentication
   echo "$REMOTECN" > $VPNDIR/TUNNEL/RemoteCN
   echo "$PSK" > $VPNDIR/TUNNEL/PSK
   echo "$GATEWAY" > $VPNDIR/Gateway
   echo "$PARAMETERS" > $VPNDIR/TUNNEL/Parameters
   echo up > $VPNDIR/STATUS
   $SCRIPTS/vpn_ctl $NAME up && $SCRIPTS/mrtg_reload
else
   exit 3
fi
```

“removevpn”;

```
#!/bin/sh
. /etc/kerbynet.conf
. $SCRIPTS/net.inc
NAME="$1"
[ -z "$NAME" ] && exit 1
CONFIG="$REGISTER/system/net/interfaces/"
VPNDIR="$CONFIG/$NAME"
if [ -r $VPNDIR/Bridge/Name ] ; then
  BR=`cat $VPNDIR/Bridge/Name`
  echo "device member of $BR (`ext_bridgename $BR`)" >&2
  exit 2
else
  if [ -r $VPNDIR/Bond/Name ] ; then
    BD=`cat $VPNDIR/Bond/Name`
    echo "device member of $BD (`ext_bondname $BD`)" >&2
    exit 2
  else
    $SCRIPTS/vpn_ctl $NAME down
    ifconfig $NAME down
        if [ "$NAME" == VPN00 ] ; then
                DEV_TYPE="tun"
        else
                DEV_TYPE="tap"
        fi
    openvpn --rmtun --dev-type $DEV_TYPE --dev $NAME
    rm -rf $VPNDIR
    $SCRIPTS/nb_vpn
  fi
fi
```

“tapcreate”;

```
#!/bin/sh
. /etc/kerbynet.conf
. $SCRIPTS/net.inc
CONFIG="$REGISTER/system/net/interfaces"
cd "$CONFIG"
INTERFACES=`ls -d VPN[0123456789][01234567890] 2> /dev/null`
for i in $INTERFACES ; do
        if [ "$i" == VPN00 ] ; then
                DEV_TYPE="tun"
```

```
        else
                DEV_TYPE="tap"
        fi
    ifconfig $i 2>/dev/null >/dev/null || openvpn --dev-type $DEV_TYPE --dev $i --mktun >/dev/null
2>/dev/null
   $SCRIPTS/setvlans $i
done
```

“vpnconfig”;

```
#!/bin/sh
. /etc/kerbynet.conf
. $SCRIPTS/net.inc
CONFIG="$REGISTER/system/net/interfaces"
export NOVPNWAIT=yes
cd "$CONFIG"
if ! [ -d VPN99 ] ; then
  mkdir VPN99
  echo up > VPN99/STATUS
  echo "Host-to-LAN OpenVPN Interface" > VPN99/Description
  VPNGW="`cat $REGISTER/system/openvpn/Gateway 2>/dev/null`"
  VPNNM="`cat $REGISTER/system/openvpn/Netmask 2>/dev/null`"
  mkdir -p VPN99/IP/00
  if [ -z "$VPNGW" ] ; then
    VPNGW=192.168.250.254
    VPNNM=255.255.255.0
  fi
  echo $VPNGW > VPN99/IP/00/IP
  echo $VPNNM > VPN99/IP/00/Netmask
  echo up > VPN99/IP/00/STATUS
fi
INTERFACES=`ls -d VPN[0123456789][01234567890] 2>/dev/null`

for i in $INTERFACES ; do
        if [ "$i" == VPN00 ] ; then
                DEV_TYPE="tun"
        else
                DEV_TYPE="tap"
        fi
   if ifconfig $i 2>/dev/null >/dev/null || openvpn --dev-type $DEV_TYPE --dev $i --mktun ; then
     $SCRIPTS/setinterface $i
#     $SCRIPTS/setvlans $i
     $SCRIPTS/vpn_ctl $i
   fi
done
```

“vpn_ctl”;

```
#!/bin/sh
. /etc/kerbynet.conf
function TERM () {
  I=0
  while [ $I -lt 30 ] ; do
    PID="`ps -ef |grep openvpn |grep " --dev $1 " |awk '{print $2}'`"
    if ! [ -z "$PID" ] ; then
        kill -TERM $PID
        sleep 1
    else
        return 0
    fi
    I=$((I+1))
  done
  PID="`ps -ef |grep openvpn |grep " --dev $1 " |awk '{print $2}'`"
  if ! [ -z "$PID" ] ; then
     kill -9 $PID
     sleep 3
  fi
  return 1
}
INTERFACE="$1"
STATUS="$2"
[ -z "$INTERFACE" ] && exit 1
CONFIG="$REGISTER/system/net/interfaces/$INTERFACE"
if [ "$INTERFACE" == VPN99 ] ; then
  exit
fi
if [ "$INTERFACE" == VPN00 ] ; then
        DEV_TYPE="tun"
```

```
else
        DEV_TYPE="tap"
fi
if cd "$CONFIG" 2> /dev/null ; then
  if [ -z "$STATUS" ] ; then
     STATUS=`cat $CONFIG/STATUS`
  fi
  if [ "$STATUS" == up ] ; then
     if TERM $INTERFACE; then
        NUM=${INTERFACE:3:2}
        if [ "${NUM:0:1}" == 0  ] ; then
          NUM=${NUM:1:1}
        fi
        MGT=$[34000+$NUM]
        REMOTEIP=`cat TUNNEL/RemoteIP`
        if ! [ -z "$REMOTEIP" ] ; then
          REMOTEIP="--remote $REMOTEIP"
        fi
        LOCALIP="`$SCRIPTS/vpn_localip $INTERFACE 2>/dev/null`"
        if ! [ -z "$LOCALIP" ] ; then
          LOCALIP="--local $LOCALIP"
        fi
       PORT=`cat TUNNEL/Port`
        PROTO=`cat TUNNEL/Proto`
        AUTHENTICATION=`cat TUNNEL/Authentication 2>/dev/null`
        TLSROLE=`cat TUNNEL/TLSRole`
        REMOTECN=`cat TUNNEL/RemoteCN`
        COMPRESSION=`cat TUNNEL/Compression`
        CRYPTO=`cat TUNNEL/Crypto`
        PARAMETERS="`cat TUNNEL/Parameters`"
        if ! [ -z "$REMOTECN" ] ; then
          REMOTECN="--tls-remote `echo $REMOTECN | sed -r 's/[# {}\(\)\^?!*[]/_/g'`"
        fi
        if [ "$TLSROLE" == Server ] ; then
           TLSROLE=server
           RESTART=7
       else
           TLSROLE=client
           RESTART=11
        fi
        if [ "$PROTO" == TCP ] ; then
           PROTO=tcp-$TLSROLE
        else
          PROTO=udp
        fi
        if [ "$COMPRESSION" == yes ] ; then
           COMPRESSION="--comp-lzo"
        else
           COMPRESSION=""
        fi
        if [ "$CRYPTO" != yes ] ; then
           CRYPTO="--cipher none"
        else
           CRYPTO=""
        fi
        if [ "$AUTHENTICATION" = PSK ] ; then
          AUTHSTRING="--secret /tmp/$INTERFACE.psk"
          echo "-----BEGIN OpenVPN Static key V1-----" > "/tmp/$INTERFACE.psk"
          cat $CONFIG/TUNNEL/PSK >> "/tmp/$INTERFACE.psk"
          echo "-----END OpenVPN Static key V1-----" >> "/tmp/$INTERFACE.psk"
          chmod 600 /tmp/$INTERFACE.psk
        else
                AUTHSTRING="--tls-$TLSROLE  --dh  $SSLDIR/dh.pem  --ca  $SSLDIR/trusted_CAs.pem  --cert
$CONFIG/TLS/cert.pem --key $CONFIG/TLS/key.pem $REMOTECN"
        fi
        if ! ps -ef |grep -w "checkvpn"|grep -q -w $INTERFACE ; then
          $SCRIPTS/checkvpn $INTERFACE &
        fi
        if ! $SCRIPTS/vpn_checknbgw $INTERFACE ; then
          GW="`cat $CONFIG/Gateway 2>/dev/null`"
          GWDESC="`cat $REGISTER/system/net/nb/Gateways/$GW/Description 2>/dev/null`"
           if [ ! -f /tmp/$INTERFACE.nogw ] ; then
             touch /tmp/$INTERFACE.nogw
             logger -t ${INTERFACE}_L2L "Unable to connect using the selected Gateway ($GWDESC)"
          else
            rm /tmp/$INTERFACE.nogw
          fi
           exit 25
        fi
        $SCRIPTS/nb_vpn
```

```
           if bash -c "openvpn --dev $INTERFACE $LOCALIP $REMOTEIP --port $PORT --proto $PROTO
$AUTHSTRING --dev-type $DEV_TYPE --float --keepalive 1 11 --script-security 3 --management 127.0.0.1
$MGT --daemon ${INTERFACE}_L2L $COMPRESSION $CRYPTO $PARAMETERS --down '$SCRIPTS/vpn_mii'" ; then
           SEM="/tmp/VPN_MII_$INTERFACE"
           rm -f $SEM
           $SCRIPTS/vpn_mii $INTERFACE
           [ -z "$NOVPNWAIT" ] && sleep 5
        else
           exit 5
        fi
     else
        exit 2
     fi
  else
     kill `ps -ef |grep -w checkvpn |grep -w $INTERFACE |awk '{print $2}'` 2>/dev/null
     TERM $INTERFACE || exit 1
     ifconfig $INTERFACE down
  fi

fi
```

“vpn_start”;

```
#!/bin/sh
. /etc/kerbynet.conf
function TERM () {
  killall -w -TERM vpn 2> /dev/null
}
ENABLED=`cat $REGISTER/system/openvpn/Enabled 2>/dev/null`
TERM
NETMASK=`cat $REGISTER/system/openvpn/Netmask 2>/dev/null`
GW=`cat $REGISTER/system/openvpn/Gateway 2>/dev/null`
if [ -n "$GW" ] ; then
   if [ "$GW" != "`cat $REGISTER/system/net/interfaces/VPN99/IP/00/IP 2>/dev/null`" -o "$NETMASK" !=
"`cat $REGISTER/system/net/interfaces/VPN99/IP/00/Netmask 2>/dev/null`" ] ; then
    if ! [ -r $REGISTER/system/net/interfaces/VPN99/Bridge/Name ] ; then
      mkdir -p $REGISTER/system/net/interfaces/VPN99/IP/00/
      echo $GW > $REGISTER/system/net/interfaces/VPN99/IP/00/IP
      echo $NETMASK > $REGISTER/system/net/interfaces/VPN99/IP/00/Netmask
      echo up > $REGISTER/system/net/interfaces/VPN99/IP/00/STATUS
      $SCRIPTS/setinterface VPN99
    fi
  fi
  $SCRIPTS/dns_hup
fi
iptables -t nat -D POSTROUTING -j OpenVPN 2>/dev/null
iptables -t nat -F OpenVPN 2>/dev/null
iptables -t nat -X OpenVPN 2>/dev/null
if [ "$ENABLED" == yes ] ; then
   $SCRIPTS/vpn_restart_x509
   PROTO=`cat $REGISTER/system/openvpn/Proto 2>/dev/null`
   PORT=`cat $REGISTER/system/openvpn/Port 2>/dev/null`
   PARAM=`cat $REGISTER/system/openvpn/Parameters 2>/dev/null`
   AUTHENTICATION=`cat $REGISTER/system/openvpn/Authentication 2>/dev/null`
   IPMIN=`cat $REGISTER/system/openvpn/IPMin 2>/dev/null`
   IPMAX=`cat $REGISTER/system/openvpn/IPMax 2>/dev/null`
   DNS=`cat $REGISTER/system/openvpn/DNS 2>/dev/null`
   NAT=`cat $REGISTER/system/openvpn/NAT 2>/dev/null`

   if [ "$AUTHENTICATION" == Password ] ; then
     NOCERTREQ="--client-cert-not-required"
   fi
   if [ "$AUTHENTICATION" != X509 ] ; then
     AUTHSCRIPT="--auth-user-pass-verify $SCRIPTS/ov_pw_auth via-env --username-as-common-name"
     PUSHAUTH="auth-user-pass"
   fi
   if [ -n "$IPMIN" -a -n "$IPMAX" ] ; then
     POOL="--ifconfig-pool $IPMIN $IPMAX $NETMASK"
     if [ "$NAT" == yes ] ; then
       iptables -t nat -N OpenVPN
       iptables -t nat -A OpenVPN -m iprange --src-range $IPMIN-$IPMAX -j MASQUERADE
       iptables -t nat -A POSTROUTING -j OpenVPN
     fi
   fi
   if [ -n "$DNS" ] ; then
     RESOLVER="dhcp-option DNS $DNS"
   fi

   if [ -n "$GW" ] ; then
```

```
      NETS=`cat $REGISTER/system/openvpn/Nets 2>/dev/null`
      PUSHGW="route-gateway $GW"
      if [ -z "$NETS" ] ; then
        REDIRECTGW="redirect-gateway"
      else
        #PUSHNETS0="route remote_host 255.255.255.255 net_gateway 1"
# modifica per evitare problemi di routing ed errori nei log.
        PUSHNETS0="route remote_host 255.255.255.255 vpn_gateway 1"
        for NET in $NETS ; do
           IPNET=`echo $NET | awk -F/ '{print $1}'`
           MASKNET=`echo $NET | awk -F/ '{print $2}'`
           if ! echo $MASKNET |grep -q '\.' ; then
             MASKNET=`$SCRIPTS/netmask $MASKNET`
           fi
           PUSHNETS="$PUSHNETS --push \"route $IPNET $MASKNET\""
        done
      fi
    fi
    MGT=34099
      bash -c "vpn --dev-type tap --dev VPN99 --mode server --tls-server --proto $PROTO --port $PORT
--dh        /etc/ssl/dh.pem        --ca        $REGISTER/system/openvpn/Auth/X509/CAFile        --cert
$REGISTER/system/openvpn/TLS/cert.pem    --key    $REGISTER/system/openvpn/TLS/key.pem    $NOCERTREQ
$AUTHSCRIPT  --daemon  VPN99_H2L  --comp-lzo  $POOL  --push  '$PUSHGW'  --push  '$REDIRECTGW'  --push
'$RESOLVER'  --push '$PUSHNETS0' $PUSHNETS  --client-connect $SCRIPTS/ov_connect --client-disconnect
$SCRIPTS/ov_disconnect  --mute  3  --management  127.0.0.1  $MGT  --keepalive  5  60  --duplicate-cn
--script-security 3 $PARAM"
fi
```

Dopo aver modificato gli script ed averli sostituiti agli originali si può con essi creare il primo ed unico tunnel VPN00 di tipo "tun".
(Volendo è possibile, facendo ulteriori modifiche agli script, costruire altri tunnel.)

Si procederà creando, ad esempio sotto la directory "/Database" di Zeroshell, una cartella che chiameremo "script_vpn" e nella quale andremo a copiare i file modificati di nostro interesse:

```
root@zeroshell root> mkdir /Database/script_vpn
root@zeroshell root> ll /Database/script_vpn/
total 28
-rw-r--r--    1 root     root             0 Feb  9 17:52 VPN00.log
-rw-r--r--    1 root     root          1231 Feb  4 12:38 createvpn
-rw-r--r--    1 root     root           663 Feb  4 12:38 removevpn
-rwxr-xr-x    1 root     root          1309 Feb  9 18:12 start_VPN00
-rw-r--r--    1 root     root           402 Feb  4 12:38 tapcreate
-rw-r--r--    1 root     root          3976 Feb  4 12:38 vpn_ctl
-rw-r--r--    1 root     root          3458 Feb  4 12:38 vpn_start
-rw-r--r--    1 root     root           926 Feb  4 12:38 vpnconfig
```

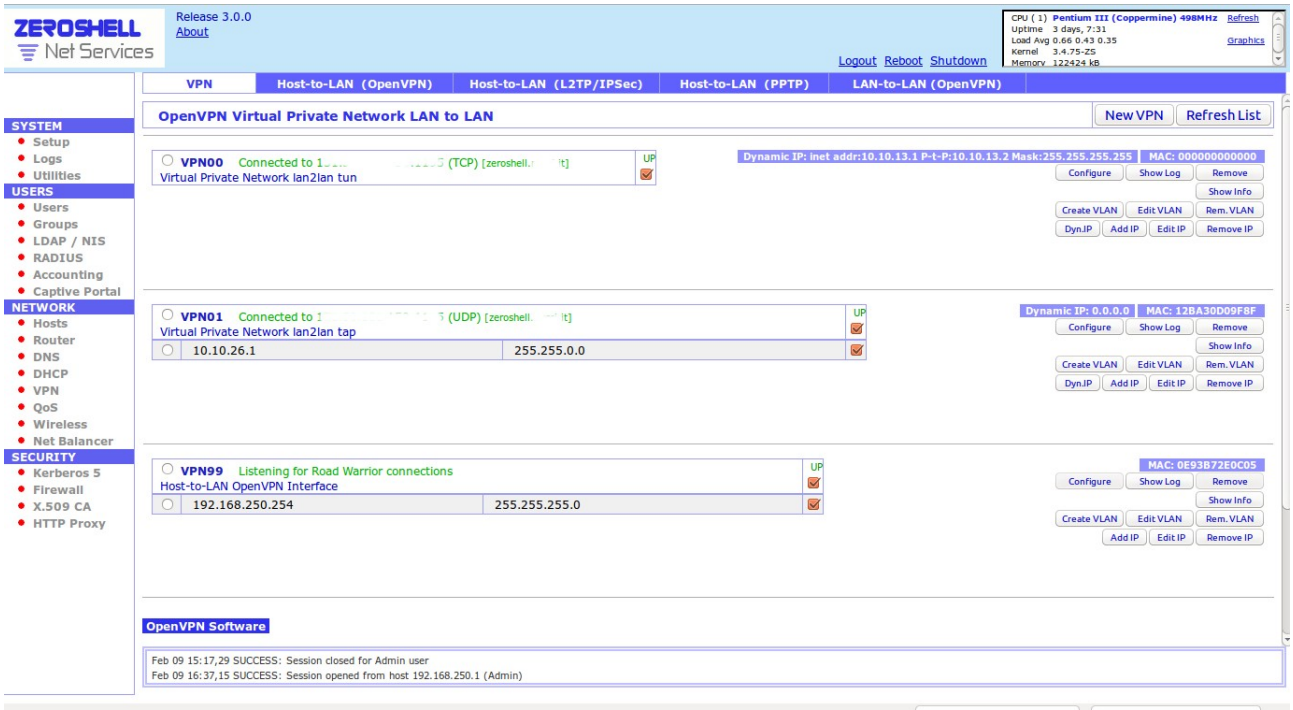
lo script "start_VPN00" è lo script da far partire come post al boot, il cui contenuto è:

```
root@zeroshell root> cat /Database/script_vpn/start_VPN00
#!/bin/sh
# Startup Script
# Si sostituiscono le copie originali con i file modificati
rm -f /root/kerbynet.cgi/scripts/createvpn
cp -f /Database/script_vpn/createvpn /root/kerbynet.cgi/scripts
rm -f /root/kerbynet.cgi/scripts/removevpn
cp -f /Database/script_vpn/removevpn /root/kerbynet.cgi/scripts
rm -f /root/kerbynet.cgi/scripts/tapcreate
cp -f /Database/script_vpn/tapcreate /root/kerbynet.cgi/scripts
rm -f /root/kerbynet.cgi/scripts/vpn_ctl
cp -f /Database/script_vpn/vpn_ctl /root/kerbynet.cgi/scripts
rm -f /root/kerbynet.cgi/scripts/vpn_start
cp -f /Database/script_vpn/vpn_start /root/kerbynet.cgi/scripts
rm -f /root/kerbynet.cgi/scripts/vpnconfig
cp -f /Database/script_vpn/vpnconfig /root/kerbynet.cgi/scripts
cd /root/kerbynet.cgi/scripts
chmod 755 createvpn  removevpn  tapcreate  vpn_ctl  vpnconfig
# Si chiude ed elimina il device "tap" di default e si crea ed apre il device di tipo "tun"
/sbin/ifconfig VPN00 down
openvpn --rmtun --dev-type tap --dev VPN00
openvpn --mktun --dev-type tun --dev VPN00
/sbin/ifconfig VPN00 up
```

la stringa da inserire in “setup-start/cron” è:

```
/Database/script_vpn/start_VPN00 & > /Database/script_vpn/VPN00.log &
```

avremo quindi, anche con il reboot del sistema, sempre la nostra configurazione Openvpn di figura, dove la OpenVPN Virtual Private Network LAN to LAN **VPN00** è con device di tipo “tun”, la OpenVPN Virtual Private Network LAN to LAN **VPN01** è con device di tipo “tap” e la OpenVPN Host-to-LAN **VPN99** continua ad essere di tipo “tap”:

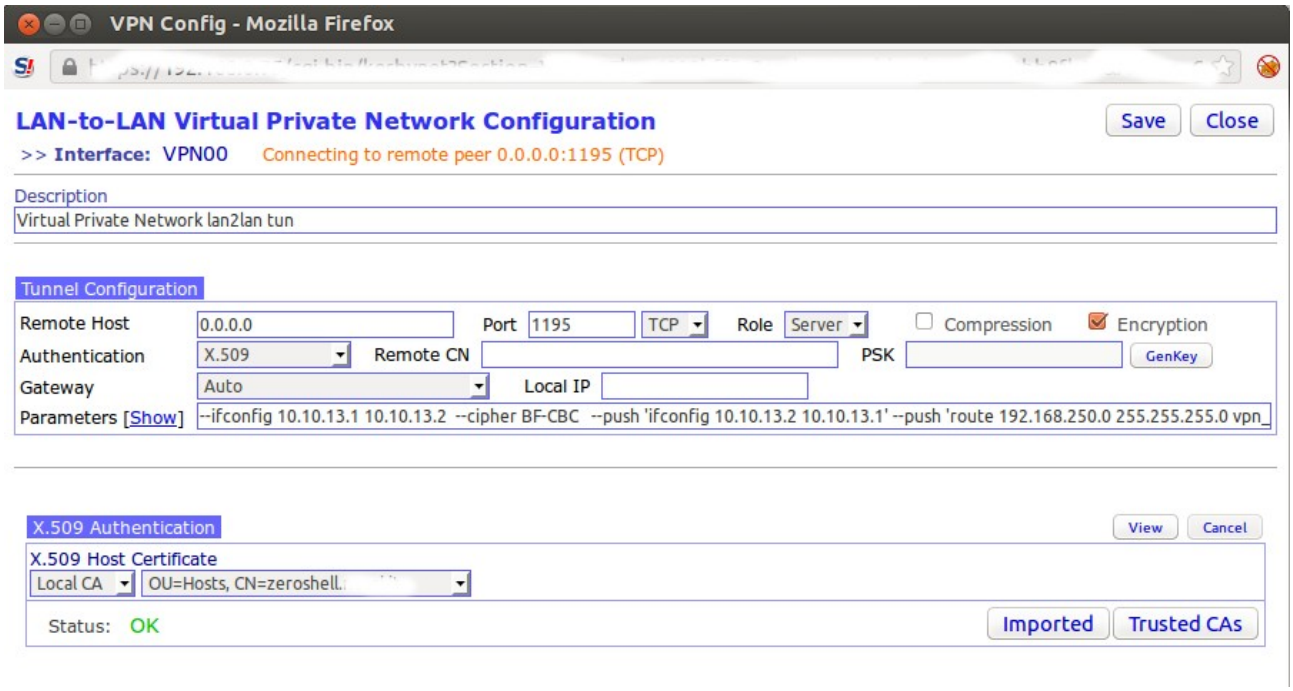


La stringa da impostare ad esempio su openvpn “zeroshell” lato server è:

```
--ifconfig 10.10.13.1 10.10.13.2 --cipher BF-CBC --push 'ifconfig 10.10.13.2 10.10.13.1' --push 'route 192.168.250.0 255.255.255.0 vpn_gateway' --push 'route 192.168.xxx.xxx 255.255.255.255 vpn_gateway' --push 'route 192.168.xxx.yyy 255.255.255.255 vpn_gateway'.
```

La figura riporta la configurazione lato server che ricalca le esperienze riportate dai vari amici nei forum:

http://www.renatomorano.net/?p=892
http://www.renatomorano.net/?p=757
http://www.zeroshell.net/forum/viewtopic.php?t=444



Invece lato client (android o ipad) si avrà:

“client.ovpn”;

```
##############################################
# 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           #
##############################################
dev tun
remote xxx.yyy.zzz.kkk
port 1195
proto tcp-client
tls-client
pull
ifconfig 10.10.13.2 10.10.13.1
cipher BF-CBC
dev-type tun
float
keepalive 1 11
script-security 3

<ca>
-----BEGIN CERTIFICATE-----
vxjsòaòòkaòkòka.......
-----END CERTIFICATE-----
</ca>
<cert>
-----BEGIN CERTIFICATE-----
sdkhhdsjsdh...........
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN RSA PRIVATE KEY-----
dfihdfihdfhvdh..............
-----END RSA PRIVATE KEY-----
</key>
```

Occorre fare ancora alcune piccole precisazioni; la configurazione esposta ha dei limiti, nel senso che si instaura una corrispondenza uno ad uno fra il client ed il server, quindi in parte si snatura la filosofia del Lan to Lan, dedicando un tunnel di ben più ampie capacità ad una sola occorrenza.
La soluzione trovata però risolve il problemino di partenza, cioè la possibilità di interconnettere alla mia lan client vpn Android e Ipad.
A discolpa si può aggiungere che non è difficile modificare le configurazioni lato server per consentire l'utilizzo in pull anche con questa topologia.

Questo è tutto!