SETTING UP OPENVPN SERVER IN UBUNTU SERVER

1. Introduction
2. Installing Packages
3. Configuration
   3.1 Copying the sample configuration file to the main openvpn folder
   3.2 Editing the Main Configuration file
4. Creating the Server Certificates
5. Creating the Client Certificates
6. Testing

1. Introduction

This SOP will demostrate on how to setup Openvpn server in a tunnel mode on an ubuntu 14.04 server.

2. Installing the packages

To install **openvpn** in a terminal enter:

```
sudo apt-get update
sudo apt-get install openvpn easy-rsa
```

3. Configuration

3.1 Copy the sample configuration file to the main openvpn folder

The example VPN server configuration file needs to be extracted to /etc/openvpn so we can incorporate it into our setup. This can be done with one command:

```
gunzip -c /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
> /etc/openvpn/server.conf
```

3.2 edit the server.conf configuration file

```
vim /etc/openvpn/server.conf
```

There are several changes to make in this file.

- Set the Server into tunnel mode, edit the line *;dev tap* to *dev tun*

```
# "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 tun0
```

- Edit dh1024.pem to say:
  dh2048.pem

```
# 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 dh1024.pem
```

This will double the RSA key length used when generating server and client keys.

- Edit Server 10.10.0.0 255.255.0.0 to much your network, point to pont ip addresses for the VPN clients will be genarated from this network.

```
# 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 10.10.0.0 255.255.0.0
```

- Add routes to be pushed to the VPN clients when connected

  in the code below, networks 192.168.80.0 255.255.255.0 and 192.168.6.0 will be pushed to the clients when connected.

```
# 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 196.216.12.0 255.255.255.0"
push "route 192.168.80.0 255.255.255.0"
push "route 192.168.6.0 255.255.255.0"
```

4. Creating Server Certificates

- First copy over the Easy-RSA generation scripts.

```
cp -r /usr/share/easy-rsa/ /etc/openvpn
```

- Next, edit `/etc/openvpn/easy-rsa/vars` adjusting the following to your environment:

```
export KEY_COUNTRY="MW"
export KEY_PROVINCE="LL"
export KEY_CITY="Lilongwe"
export KEY_ORG="Baobab Health Trust"
export KEY_EMAIL="baobab@baobabhealth.org"
```

- Now Create the certificates by entering the below commands

```
cd /etc/openvpn/easy-rsa/
source vars
./clean-all
./build-dh
./pkitool --initca
./pkitool --server server
cd keys
openvpn --genkey --secret ta.key
sudo cp server.crt server.key ca.crt dh1024.pem ta.key 
/etc/openvpn/
```

5. Creating Client Certificates

- The VPN client will also need a certificate to authenticate itself to the server. To create the certificate, enter the following in a terminal:

```
cd /etc/openvpn/easy-rsa/
source vars
./pkitool hostname
```

Replace *hostname* with the actual hostname of the machine connecting to the VPN

- Copy the following files to the client:

/etc/openvpn/ca.crt

/etc/openvpn/easy-rsa/keys/hostname.crt

/etc/openvpn/easy-rsa/keys/hostname.key

/etc/openvpn/ta.key

Remember to adjust the above file names for your client machine's *hostname***.**

It is best to use a secure method to copy the certificate and key files. The **scp** utility is a good choice, but copying the files to removable media then to the client, also works well.

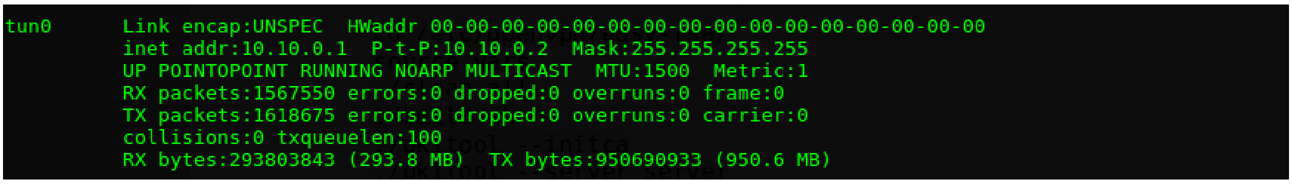
6. Testing

- Restart the openvpn server

  /etc/init.d/openvpn restart

- After restarting the server check if the tunnel interface has come up

  type ifconfig and see if an interface like the one shown below is up



```
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.10.0.1  P-t-P:10.10.0.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:1567550 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1618675 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:293803843 (293.8 MB)  TX bytes:950690933 (950.6 MB)
```

- fot testing on the client side refer to the Setting up openvpn client on an ubuntu server ver 0.0.2