Provisioning yealink phones

to connect with openvpn

## Date: August 10th 2017

## Author: Truvis Thornton

## Website: <https://www.linkedin.com/in/truvisthornton>

**\*\* WORK IN PROGRESS \*\***

The goal of this document is to provide more of a step-by-step setup and basic introduction for people who do not have a firm grasp with working with VPNs, firewalls and other such services and protocols. If you have any questions please feel free to reach out to me if you would like a more indepth understanding as to why the configuration and setup is done as poste in this document.

This document does not cover server security or configuration. The scope of this document is stricly to create a working openVPN tunnel for YeaLink phones. If you have little to know Linux Server experience, you should seekout a professional for properly securing and setting up.

# openvpn server setup

## LINUX Server:

Your Linux server MUST support tun in order for this to work. Most Linux kernels will have this compiled in by default or as a module. You can test with the following:

root@truvis-test-server /e/o/c/phone# cat /dev/net/tun

cat: /dev/net/tun: File descriptor in bad state

root@ truvis-test-server /e/o/c/phone# lsmod | grep tun

tun 28672 2

If you do not get the results as shown above, you will need to resolve those issues which is outside the scope of this document. If you are on an OpenVZ kernel, this generally is not enabled by default but your hosting provider should be able to enable this support for you.

Debian/Ubuntu based install

You can begin the installation with the following:

apt-get install openvpn easy-rsa –y

This will install all the needed packages and the services.

CentOS/Fedora/RHEL based install

After the service and been installed and setup, we now need to configure firewall rulesets if there is a firewall running.

If your server is running iptables and not UFW use the following ruleset.

iptables -A INPUT -i eth0 -m state --state NEW -p udp --dport 1194 -j ACCEPT

iptables -t nat -A POSTROUTING -s 10.90.10.0/24 -o eth0 -j MASQUERADE

iptables -A OUTPUT -o tun+ -j ACCEPT

iptables -A FORWARD -i eth0 -o tun+ -m state --state RELATED,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i tun+ -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT

iptables -A INPUT -i tun+ -j ACCEPT

iptables-save

This will allow traffic to be forwarded through the device interfaces properly so that traffic betweent he two networks flow properly.

## PFSENSE FIREWALL:

Text

# phone/client setup

## T2x (does not include t27, t29):

You can use the following format for vpn.cnf

client

setenv SERVER\_POLL\_TIMEOUT 4

ns-cert-type server

reneg-sec 604800

sndbuf 100000

rcvbuf 100000

auth-retry nointeract

dev tun

dev-type tun

remote REMOTE SERVER IP 1194 udp

remote-cert-tls server

resolv-retry infinite

nobind

cipher BF-CBC

auth SHA1

tls-client

comp-lzo no

verb 3

ca /config/openvpn/keys/ca.crt

cert /config/openvpn/keys/client.crt

key /config/openvpn/keys/client.key

## T4x ():

Text

## Putting the vpn together:

You should download 7-zip

# provisionion

## NDP Server:

network.vpn\_enable="1"

openvpn.url="http://XXX.TLD/openvpn.tar"

## Manually:

M