Com.X
The complete telephony solution in a

TECHNICAL NOTE

Multi-site VPN deployment

A client has two Com.X systems at separate sites. An IAX trunk is used for traffic between the two Com.X's. The client would like to be able to transfer calls between the two sites. Furthermore, the receptionist, who is located at the head-office, would like to know the status of the extensions at the branch Com.X so that she can avoid transfers to busy users.

The client's network edge devices do not support VPN configuration. The Receptionist uses a Yealink T.28 handset with the Exp38 expansion module.

Network design:

There are a few difficulties to overcome in the above scenario. The geographical separation of the two sites means that there is no physical LAN cable between them, so any networking has to be done over the internet. We also need the T.28 reception extension to somehow get access to the branch Com.X to know whether the branch extensions are busy or not.

Since SIP info messages cannot be routed along the IAX trunk, the best approach would be for the T.28 phone to register its second account as an extension on the remote Com.X . However, the above would necessitate having port 5060 of the branch Com.X accessible from the internet, which presents security issues.

As a result, the head-office PBX needs to act as the host server of a VPN, of which the branch Com.X is a client. The reception phone then registers its first account with the head-office Com.X directly and registers its second account with the branch Com.X through the VPN, using the head-office Com.X as a routing gateway.

The above scenario was simulated at the Far South Networks labs using two Com.X1s, some test handsets simulating extensions at both sites, and one Yealink T.28 with T38 expansion module, as the reception handset. The devices were configured with the following network settings:

Device	Local IP	VPN IP
Head-office Com.X:	192.168.0.25	10.8.0.1
Branch Com.X:	192.168.0.118	10.8.0.6
Reception Handset:	192.168.0.123	

Note: Due to IP routing requirements, it is necessary for the reception phone to have a static IP and gateway.

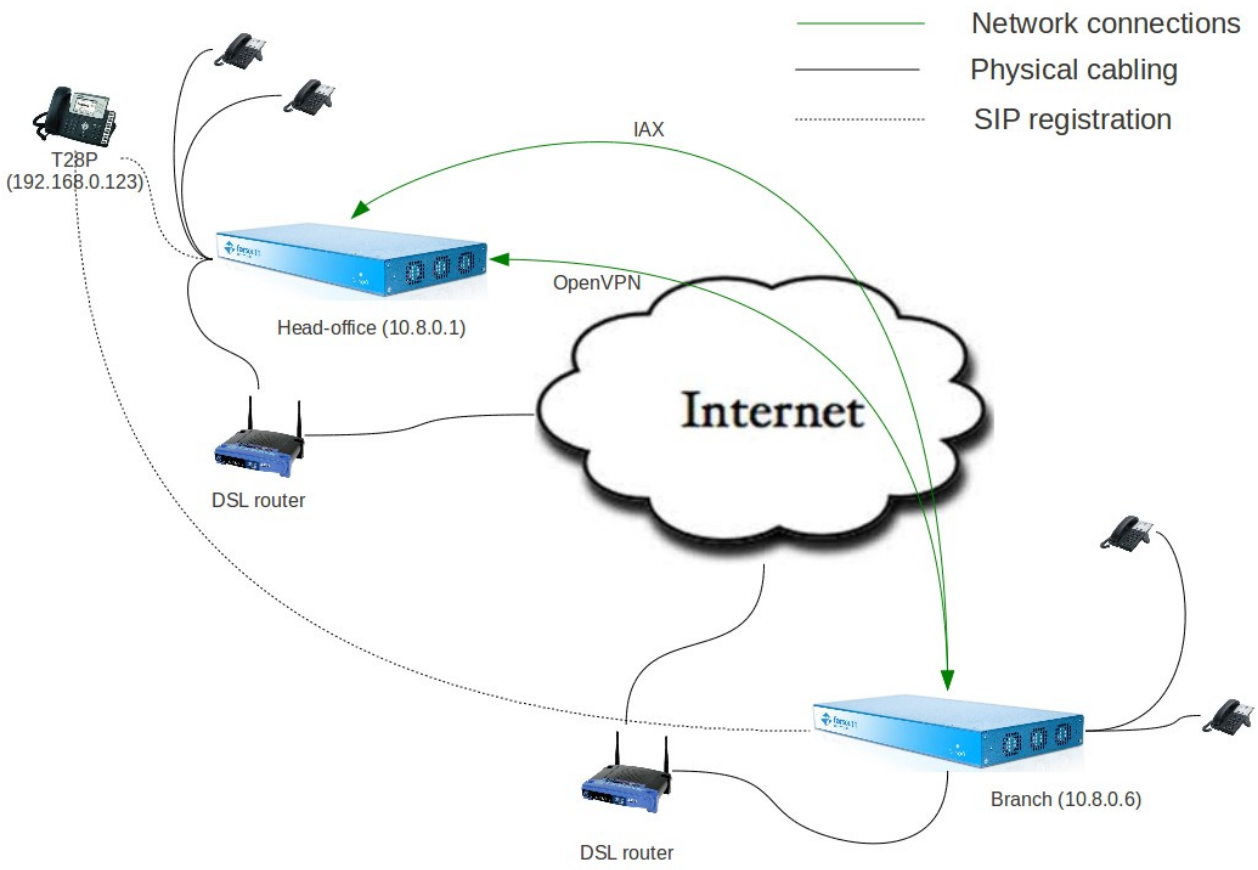

Figure 1: Network diagram

VPN Configuration

To set up a VPN, the necessary software needs to be installed on both systems and then configured as required. We will first focus on installing the software on the host server, which in this case is the head-office Com.X, which has the local IP 192.168.0.25 (you should substitute this with your own head-office system's IP in the configuration below)

In order to ensure secure VPN connectivity, the two Com.X systems need to be convinced that they are in fact connecting to their intended peer. This is accomplished by providing both systems with signed certificates that affirm their identity. These certificates are signed by a certificate authority that is installed on the head-office system. This certificate authority itself needs a certificate that the Com.X systems can use to check their peer's certificate's signature against.

Additionally, in order to encrypt the communication over the VPN, all parties (the certificate authority, the head-office Com.X and the branch Com.X) all need encryption keys. This allows the two peers to encrypt and un-encrypt communication over the VPN.

The steps below detail the creation and configuration of these required VPN components:

Head-office Configuration (VPN host)

The following commands are entered on the Com.X shell:

```
ssh comma@192.168.0.25
```

Enter the correct password and proceed as superuser:

```
sudo bash
```

Edit `/etc/apt/sources.list`

```
sudo aptitude update

sudo aptitude install openvpn dnsmasq openssl
```

Make a directory in `/etc/` to store all of the openVPN files and move examples of the required configuration files to that directory:

```
mkdir /etc/openvpn
cp /usr/share/doc/openvpn/examples/easy-rsa/2.0/* /etc/openvpn/
cd /etc/openvpn/
mkdir keys
```

`jed vars` (scroll to the bottom and enter your details, then save)

Source the `vars` script, and then build the certificate authority certificate, the server certificate and the client certificate.
Note: the names *`server-name`* and *`client-name`* are place-holders for names you can choose.

```
source ./vars
./clean-all
./build-ca
```

The commands above create the certificate authority's certificate and key. The certificate authority “signs” the head-office and branch Com.X certificates to ensure they are valid.

```
touch /etc/openvpn/keys/index.txt
echo 01 > /etc/openvpn/keys/serial
./build-key-server server-name
```

The commands above create a certificate and key for the head-office VPN server to allow it to reliably identify itself to the branch Com.X

```
./build-key client-name
```

The command above create a certificate and key for the branch VPN client to allow it to reliably identify itself to the head-office Com.X

```
./build-dh
```

The command above builds the necessary encryption parameters to ensure strong encryption communication.

Copy the `client-name.key, client-name.crt` and `ca.crt` files to the VPN client (the branch Com.X):

```
scp keys/client-name.key comma@192.168.0.118:~/
scp keys/client-name.crt comma@192.168.0.118:~/
scp keys/ca.crt comma@192.168.0.118:~/
```

Copy some example configuration files to your `/etc/openvpn` folder and edit the `server.conf` file once it has been extracted:

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

Enter the local IP of the machine with netmask (e.g. 192.168.0.25/24 – be specific, 192.168.0.0/24 won't work)
Change the protocol to tcp
Point to the correct certificate and key files in `/etc/openvpn`:

```
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/server-name.crt
key /etc/openvpn/keys/server-name.key
```

Replace the default `dh1024.pem` entry with:

```
dh /etc/openvpn/keys/dh1024.pem
```

Configure the desired VPN server IP and IP Routes:

```
server 10.8.0.0 255.255.255.0
push "route 192.168.0.0 255.255.255.0"
```
(i.e. to allow LAN access)

Uncomment:
```
user nobody
group nogroup
```
(You may need to change “nobody” to “nogroup”)

Enable the VPN server to automatically start on system startup, and ensure that the level of detail reported will be helpful:

`jed /etc/default/openvpn` and enable `(AUTOSTART="all")`
`jed /etc/openvpn/server.conf` and set `'verb 5'` for logging

You can track the progress of the VPN server as it starts up the tail command:

```
tail -f /var/syslog
```

You can also try the server out as a foreground process and see if it is working:

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

The above line will run openVPN in the foreground and output any logging and errors to the terminal. If there are no errors, you will be able to stop the command (`ctrl-c`) and run the openVPN service in the background:

```
sudo /etc/init.d/openvpn start
```

Congratulations! You now have openVPN running on your head-office Com.X. Now on to the client.

Branch configuration (VPN client)

As with the host, we first need to install the correct software packages on to the client Com.X. First edit `/etc/apt/sources.list` to include the necessary repositories.

Then, from the terminal:

```
sudo aptitude update
sudo aptitude install openvpn dnsmasq openssl network-manager-openvpn
```

`jed /etc/default/openvpn` and set `AUTOSTART="all"` and `'verb 5'`

Open the syslog in a new terminal window, to view progress in real time. This is most useful for troubleshooting, should there be any errors:

```
tail -f /var/syslog
```

Ensure that the client certificate and key files created earlier and copied to the comma user's home directory are moved somewhere safe. For illustrative purposes below we refer to `/home/comma/secure/`

Then create the client.conf file:

```
sudo jed /etc/openvpn/client.conf
```

and edit it to include the lines below:

```
dev tun
client
proto tcp
remote 192.168.0.25 1194   (Use your server IP here)
resolv-retry infinite
nobind
user nobody
group nogroup
keepalive 30 120
# Try to preserve some state across restarts.
persist-key
persist-tun
ca /home/comma/secure/ca.crt
cert /home/comma/secure/client-name.crt
key /home/comma/secure/client-name.key
comp-lzo

# Set log file verbosity.
Verb 5
```

The above completes the client configuration, and you should now be able to start your openVPN service:

```
sudo /etc/init.d/openvpn start
```

Configuring Asterisk to accept traffic from the VPN

In order for Asterisk to allow traffic from the VPN subnet, the following line needs to be added to `/etc/asterisk/sip_general_custom.conf` on both the head-office and branch Com.X systems.

```
localnet = 10.8.0.5/255.255.255.0
```

Testing your VPN

The steps below should help in confirming / diagnosing the VPN configuration and connectivity:

Use `ifconfig` to show the VPN tun0 device and subnet.
Use `ip route` to show the system's network routes through the standard network as well as the VPN tunnel device.

If the route openVPN configured clashes with your normal default route, some additional routing configuration may be needed to get routing working. Start by deleting the openVPN default route (usually a duplicate route for the local network).

If your VPN and IP routes have been set up correctly, you should be able to ping each Com.X across the VPN, and should also be able to ping devices on either network from the appropriate Com.X.

i.e. in the above scenario, the head-office Com.X should be able to ping the branch Com.X at 10.8.0.6 and the branch Com.X should be able to ping the head-office Com.X at 10.8.0.1.

TECHNICAL NOTE

Configuring the Phone to register on both systems

Once you have a working VPN, you will need to configure the reception phone to register with both Com.Xs, with the registration to the branch Com.X being done through the head-office Com.X, which acts as a gateway. The reception handset needs to have a static IP for the desired behaviour to be achieved.

Head-office Com.X configuration

We first need to configure the head-office Com.X to act as an ipv4 gateway. This is done by editing `/etc/sysctl.conf` to contain the line `net.ipv4.ip_forward = 1.` You will most likely need only to change the default "0" to a "1". Reboot the system after making this change to activate the routing functionality.

Phone Configuration

The phone needs to be configured to have one account registered on the head-office Com.X, 192.168.0.25.

The phone also needs to register at the branch Com.X. To do this, the head-office Com.X (192.168.0.25) needs to be entered as the phones default gateway, and an account registered to 10.8.0.6, the branch Com.X.

Branch Com.X configuration

Finally, a route needs to be added to the branch Com.X so that it is able to send information back to the phone, via the head-office Com.X.

```
sudo ip route add 192.168.0.123 via 10.8.0.5
```

Sample IP route tables

After successful configuration of the steps above, your routing tables should look like the reference systems' routing tables below:

Head-office Com.X (192.168.0.25 & 10.8.0.1)

10.8.0.2 dev tun0 proto kernel scope link src 10.8.0.1
192.168.101.0/24 dev eth1 proto kernel scope link src 192.168.101.1
192.168.102.0/24 dev eth2 proto kernel scope link src 192.168.102.1
192.168.103.0/24 dev eth3 proto kernel scope link src 192.168.103.1
10.8.0.0/24 via 10.8.0.2 dev tun0
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.25
default via 192.168.0.1 dev eth0 metric 100 .

Branch Com.X (192.168.0.118 & 10.8.0.6)

10.8.0.5 dev tun0 proto kernel scope link src 10.8.0.6
10.8.0.1 via 10.8.0.5 dev tun0
192.168.0.102 via 10.8.0.5 dev tun0
192.168.101.0/24 dev eth1 proto kernel scope link src 192.168.101.1
192.168.102.0/24 dev eth2 proto kernel scope link src 192.168.102.1
192.168.103.0/24 dev eth3 proto kernel scope link src 192.168.103.1
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.118
default via 192.168.0.1 dev eth0 metric 100