This instruction is for setting up a single user OpenVPN network on the Hamvoip image. VPN's or virtual private networks allow for the routing of inbound and outbound traffic through an external server, bypassing firewalls and other methods of routing that would otherwise cause issues with connections. VPN's also add a layer of security and privacy, allowing safe use in public environments.

First, you will need to set up a VDS, or virtual dedicated server. These are virtualized computer environments in a datacenter using part of a dedicated server. We also recommend downloading Bitvise as your SSH/FTP client.

For this demonstration, we will be using VirMach (http://www.virmach.com/) and their KVMLite Value+ package, which is $2.25/month. The Value+ package uses 512MB RAM and a single shared Intel CPU core at 2GHz. However, any VDS with a TUN device enabled will work. Some providers require you to request the TUN device to be enabled. We will be using Ubuntu 18.04 for the install. Also, we will assume that you are already using a Hamvoip node with the latest updates installed.

We will start on the VPN install first. Once you have logged in as root, check for updates.

```
sudo apt-get update
sudo apt-get upgrade
```

Depending on how fast your VDS is, this may take a few minutes. You will need to install iptables-persistent, which will keep your port forwarding settings after a reboot.

```
sudo apt-get install iptables-persistent
```

You will be asked to save the current rules, select Yes for both. Now, we will install OpenVPN using the openvpn-install script by Angristan on Github (https://github.com/angristan/openvpn-install) however some machines will need Curl installed to download the script. If Curl is already installed, skip this step.

```
sudo apt-get install curl
```

First, we will download the script, make it executable, and run it. Make sure you are in the root directory before you start.

```
curl -O https://raw.githubusercontent.com/angristan/openvpn-install/master/openvpn-install.sh
chmod +x openvpn-install.sh
./openvpn-install.sh
```

Follow the on-screen instructions to install OpenVPN. The script should auto-detect your public IP, if it doesn't, enter manually. IPv6 is not required, so enter “n” when prompted. Next, you will be asked to select a port. You can either use the default, pick a custom port, or random. (I use Random). Next, you will be asked for which protocol you want to use. Typically you will want to choose UDP*

* Sometimes, TCP based traffic (Supermon, SSH, etc) passing over a UDP VPN is affected when using certain cellular connections. You can try UDP, but if you end up having issues connecting via SSH or Supermon, you can change the protocol to TCP. More on this later

Next, you will be asked which DNS resolvers you want to use. This is entirely personal opinion, so I usually stick with the default, Cloudflare (3). You can skip Compression, it is not necessary. If you are not familiar with OpenVPN, use the default encryption settings (enter “N”)

Once this is all done, the script will install OpenVPN with your desired settings. This may take some time. Once done, it will prompt you to create a client configuration. For the client name, you can simply use your callsign or use whatever name you want. Next, it will ask to password protect the config.. this is not really necessary so select option 1. The config will be loaded into your root directory. You can use your FTP client to go ahead and copy the config to your PC, or just leave it there. We will come back to it later.

Next, we will set up port forwarding for your OpenVPN client. We will use iptables for this. There are other methods, such as using ufw, if you are more familiar with it. The ports we will need to open and their defaults are:

Allstar, 4569 UDP (Default, you can change this if you wish)
SSH, 222 TCP
Supermon, 80 TCP
Echolink, 5198 & 5199 UDP, 5200 TCP

The openvpn-install script should have also implemented rules for the VPN to access as well. You should not have to do anything for this. Copy the following into Notepad, replace “your.public.ip.address” with the public IP of your server. Due to the PDF formatting, you will also have to reduce the amount of lines so that each iptables command is a single line. By default, your client should use 10.8.0.2 as it's virtual IP on the VPN, so leave this for a single client set up. After done, copy the entire selection, then right click on your SSH window, and it will paste the entire selection.

```
iptables -t nat -A PREROUTING -d your.public.ip.address/32 -i eth0 -p tcp -m tcp --dport 80 -j
DNAT --to-destination 10.8.0.2:80
iptables -t nat -A PREROUTING -d your.public.ip.address/32 -i eth0 -p tcp -m tcp --dport 222 -j
DNAT --to-destination 10.8.0.2:222
iptables -t nat -A PREROUTING -d your.public.ip.address/32 -i eth0 -p udp -m udp --dport 4569
-j DNAT --to-destination 10.8.0.2:4569
iptables -t nat -A PREROUTING -d your.public.ip.address/32 -i eth0 -p udp -m udp --dport 5198
-j DNAT --to-destination 10.8.0.2:5198
iptables -t nat -A PREROUTING -d your.public.ip.address/32 -i eth0 -p udp -m udp --dport 5199
-j DNAT --to-destination 10.8.0.2:5199
iptables -t nat -A PREROUTING -d your.public.ip.address/32 -i eth0 -p tcp -m tcp --dport 5200 -j
DNAT --to-destination 10.8.0.2:5200
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 ! -d 10.8.0.0/24 -j SNAT --to-source
your.public.ip.address
iptables -t nat -A POSTROUTING -s 10.8.0.2/32 ! -d 10.8.0.2/32 -j SNAT --to-source
your.public.ip.address
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
```

Note: If you want to keep your Supermon or TCP access hidden from the default port, to prevent possible attacks, you can change the –dport to whatever port you want to use. Just remember that you will need to use this port instead of the default ports on your Hamvoip node.

When done, enter the following commands to save the port forwarding configuration. This ensures the configuration will save even if the VDS is restarted.

```
iptables-save
netfilter-persistent save
```

At this point, your OpenVPN server is ready to go. Connect to your Hamvoip node via SSH. If you have not done so recently, run menu option 1 for a system update.

Select option 9 (Start Bash shell interface) and install the Hamvoip-openvpn package.

```
pacman -R openvpn
pacman -Sy hamvoip-openvpn
```

This will install the latest OpenVPN setup for your node. Next, you will need to enable the VPN in your allstar.env file.

```
nano /usr/local/etc/allstar.env
```

Look for **export VPN_Network=”disabled”** and changed to **“enabled”** then hit **Ctrl+X** and hit **Y** to save.

Next, you will need to import your config file generated by the server. If you have not done so yet, copy the file from the server to your computer (It will be located at /root/xxyxxx.ovpn) then import it to your node to the /etc/openvpn folder. There are two methods to importing your config that you can do. You can either delete the default client.conf, import your .ovpn client configuration and rename it client.conf or you can import the .ovpn file, give it a custom name (i.e. yourcallsign.conf) then change it in rc.allstar. Once imported, you will need to comment out one line in the config.

```
nano /etc/allstar/yourconfig.conf
```

Look for the line “explicit-exit-notify” and add a # to the beginning to comment it out.

If you change the name of your config to something other than client.conf, you will need to edit a line in rc.allstar for it to recognize your config file.

```
nano /usr/local/etc/rc.allstar
```

You will need to a line starting with “**if [“${VPN_NETWORK,,}” == “enabled”]**” and replace the client.conf argument with the client file name you chose. When finished, **Ctrl+X**, then hit **Y** to save.

Once done, restart your node, either from the panel or entering **sudo reboot** in Bash. If everything is done correctly, when your node boots up, it should connect to the VPN and you will see the VPN's IP at the top (10.8.0.2, your.LAN.IP.here). And success, you now have a working VPN connection. Test all your connections (inbound and outbound Allstar, Echolink, Supermon, SSH, etc)

With some cellular providers, using UDP may cause issues with connecting to Supermon/SSH, or with the VPN disconnecting randomly. If this happens, modify both your client & server configuration files (located in /etc/openvpn) from “proto udp” to “proto tcp” and restart both.

A big shout out to David McGough KB4FXC for his help with setting up OpenVPN and his work on the Hamvoip-openvpn package, and a shout out to Rob Seaman VK6LD for some of his suggestions in the ARM-Allstar list.

Mike Sullivan, KN4IMU
kn4imu@gmail.com

Modified 09/15/2020