Setting Up VPN's Using a Raspberry PI

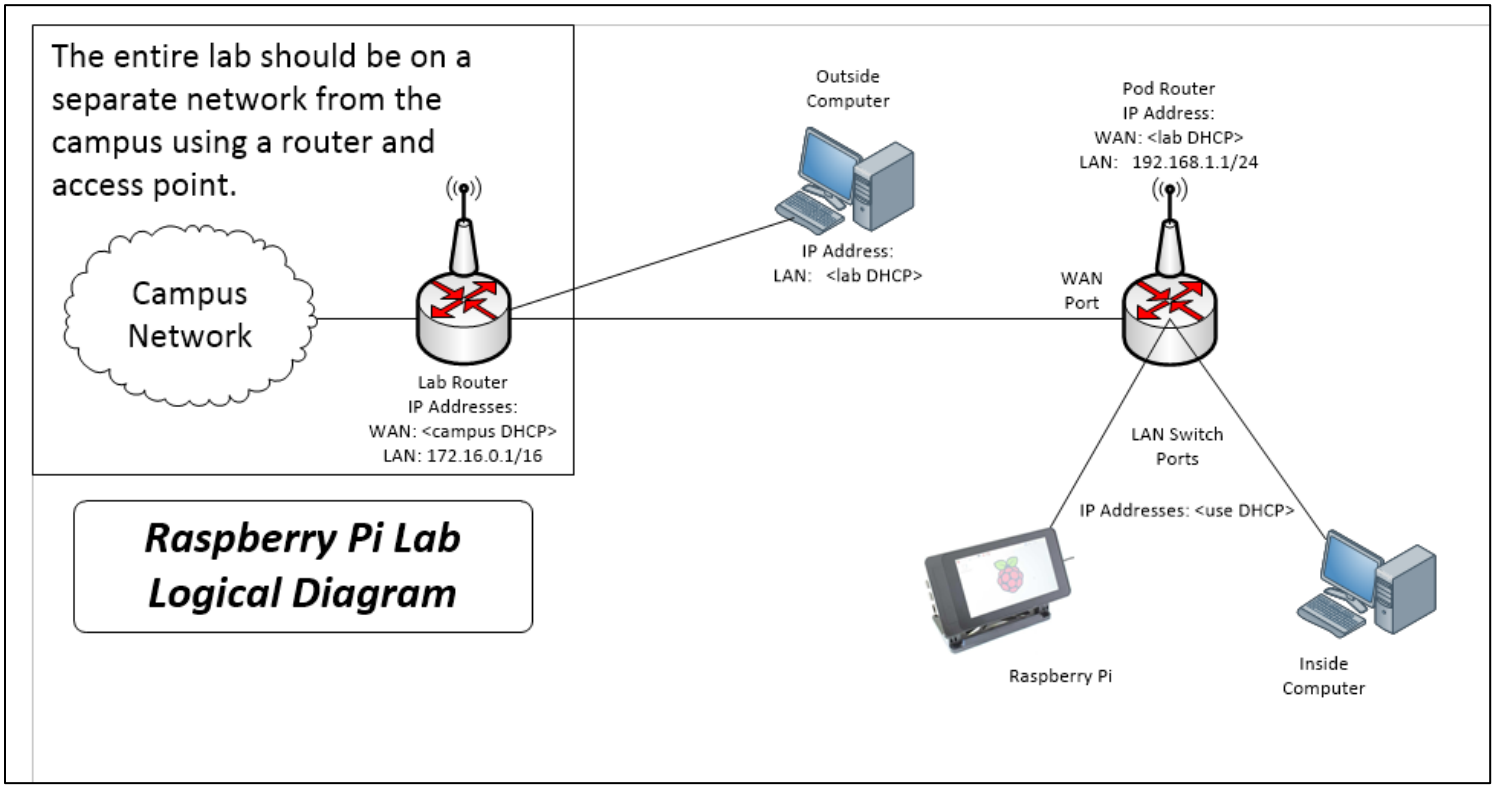
Much of this material was taken from "dayz" website at: [Kamil's Lab - How To Turn Your Raspberry Pi Into a Home Server Using PiVPN](). It's a great site for a whole bunch of Pi related labs. Please visit.

Environment Setup

You will need the following:

- Raspberry Pi (you can use either a "headless" Pi or one with the 7" built in display)
- SD Card with the current version of Raspbian Lite (the version without the GUI)
- Two computers (laptop or desktop)
  - One will be on the "inside" to configure the PI and the router
  - One will be on the "outside" to connect to the OpenVPN server
- Small Office/Home Office Router/Switch/Access Point (DLink, Linksys, Netgear, etc) to connect the Raspberry Pi and computers to the "outside"
  These instructions assume you'll be using DD-WRT, but the port forwarding instructions are basically the same on most of the routers.
- CAT5 cables to run between the Pi, the laptop and the router
- If you are using the Pi with the 7" display, you will also need a keyboard

Below is the logical diagram for the lab pod.



1. Connect the Inside computer to one of the switch ports on the router.

2. Configure the Router/Switch/Access Point with the following IP addresses:
   WAN-Side: Use **DHCP** *which it will get from the lab's DHCP server*
   LAN-Side: Use **192.168.1.1** as the router's IP address
   Make sure the router is set to provide IP addresses using DHCP
   It is important to check the DHCP range that is configured because you will need to manually set a Static IP address on the Raspberry Pi. Make sure the starting address is ***NOT 192.168.1.2*** *it should be something like* ***192.168.1.50*** *or* ***192.168.1.100*** *the number of addresses or the last address is not critical*

3. Image the Raspberry Pi OS (Rasbian) onto the MicroSD card. Use the "Lite" version (no GUI).

4. Connect the Raspberry Pi to one of the switch ports on the router.
   If you have built the Raspberry Pi using the Smart Pi Touch case with the 7 inch touchscreen, then you will need to connect a USB Keyboard. If you are using the Raspberry Pi headless, then you will need to use SSH to connect.

5. Connect the Outside computer to the lab network

6. Power up the Raspberry Pi by plugging it into the power supply.

Part 1 - Installing the PiVPN Implementation of the OpenVPN Protocol

OpenVPN is an open source implementation of an SSL-based VPN. It is a client/server foundation that requires SSL certificates (public/private key infrastructure). Since it is open source, the server and clients can be implemented on many devices, such as routers, servers and virtual appliances. **OpenVPN** Technologies has a simple to implement appliance that can be installed on bare metal or run as a VM. The server software is free, but only allows for two concurrent connections. If you want more than two concurrent connections, you must purchase additional licenses. Third-party implementations do not have this restriction.

PiVPN is a bash-script that takes care of much of the heavy lifting that would be required to install **OpenVPN** on a Raspberry Pi. It makes life pretty easy.

Installation of **PiVPN** (The software we will be using as our VPN server) is a breeze. You simply have to run just one command to install **PiVPN**. This assumes you already have a Raspberry PI OS up and running. You only need the lite version. If you are using the Pi with an attached monitor, keyboard and mouse you could use the full version with the GUI … the setup is the same.

> *If you do not have a monitor and keyboard, then you can use an IP scanner, such as* **IPScan** *to find the IP address from a computer attached to the same network in order to connect to the Pi using SSH.*
>
> *Use a terminal program, such as* **PuTTY** *to* **SSH** *into the Raspberry Pi.*

- Login to the Raspberry Pi, the default user name is **pi** and use **raspberry** as the password.

- Update Raspberry Pi OS (Rasbian) …

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

  … Patience, this can take quite a bit of time.

Set the correct keyboard and location information (the default is a UK keyboard and location)

- At the command prompt type: **`sudo raspi-config`**

- On the following screens, do the following:

    - From the menu options, choose option **4: Localisation Options**
    - From the next menu, choose option **I1: Change Locale**
    - From the list of locales, use the **DOWN-ARROW** key to:
        … deselect: **en_GB.UTF-8 UTF-8** … by pressing the **SPACE** bar
        … choose: **en_US.UTF-8 UTF-8** … by pressing the **SPACE** bar
    - **TAB** down to **OK** and press **ENTER**
    - Use the **DOWN-ARROW** key to select **en_US.UTF-8**
    - **TAB** down to **OK** and press **ENTER**
    - On **DEFAULT-LOCALS** select **NONE**, then **TAB** down to **OK** and press **ENTER**
    - You will be returned to the Configuration Menu

    - From the menu options, choose option **4: Localisation Options**
    - From the next menu, choose option **I3: Change Keyboard Layout**
    - Use the **UP ARROW** key to select: **Generic 104-key PC** and press **ENTER**
    - On the next screen, Keyboard Layout, use the **DOWN ARROW** key to select **OTHER** and press **ENTER**
    - On the next screen, **Country of Origin,** use the **DOWN ARROW** key to select **English (US)** and press **ENTER**
    - On the next screen, **Keyboard Layout**, use the **UP ARROW** key to select **English (US)** and press **ENTER**
    - On the next screen, **Keyboard to Function**, make sure **The Default for the Keyboard Layout** is selected and press **ENTER**
    - On the **Compose Key** screen, **,** make sure **No Compose Key** is selected and press **ENTER**
    - Press **TAB** to **FINISH** and press **ENTER**

You will now be pulling the script file off of the web and piping it into a bash script

```
sudo curl –L http://install.pivpn.io | bash
```

> *Just a quick side-note, running a command like this is dangerous. Basically what the command is doing is going to http://install.pivpn.io and parsing the data then running it in the command line. If you run a similar command from an untrusted source you can do some damage and it is very dangerous to do so. You can type https://install.pivpn.io in your browser to see the exact commands being run.*

After you run the command above you should get the PiVPN Automated Installer welcome screen.
Press **ENTER** on **OK** to continue:

- The next screen is informing you that the OpenVPN server requires a static address. The default is to use the DHCP address assigned to PiVPN by the router. This screen lets you know that you can continue to use the DHCP address assigned by the router, or you can change it to a Static IP address
  Press **ENTER** on **OK** to continue:

- The next screen asks if you wish to use the assigned DHCP address or to change it to a Static IP address (that is out of the DCHP range).
  Make sure **NO** is selected (if it is not, use the **TAB** key to select it) and press **ENTER**.

- The next screen will ask if you wish to use the currently assigned (through DHCP) address or to change it.
  TAB to **NO** and press **ENTER**.

- The next screen asks for the desired IPv4 address.
  Change the address to **192.168.1.2/24** (this might be different depending on the DHCP range).
  TAB to **OK** and press **ENTER**.

- The next screen asks for the IPv4 Default Gateway.
  The value should be **192.168.1.1** … if it is, **TAB** to **OK** and press **ENTER**
  If it is not, then change it to **192.168.1.1**, then **TAB** to **OK** and press **ENTER**

- The next screen asks you to confirm the settings. Check to make sure that the correct addresses are being used.
  Press **ENTER** on **YES**

- The next screen will ask you to choose a local user that will hold the OVPN configurations.
  Press **ENTER** on **OK**

- The next step will ask you to pick a user that will have the PiVPN configuration settings. There is only one user created by the installation script which is **pi**.
  Tab to **OK** and press **ENTER**

- The next screen is asking what Installation Mode you would like to use. You have two options, WireGuard or OpenVPN. We will be installing OpenVPN, it is the traditional, flexible and more trusted VPN protocol with additional features.
  Use the **DOWN ARROW** and press the **SPACE** bar to select **OpenVPN**,
  then **TAB** to **OK** and press **ENTER**

- The next screen shows the settings that PiVPN recommends. For this lab, we will use the recommended settings.
  Make sure **NO** is selected and press **ENTER**

- The screen will allow you to pick a Port for the VPN connections. The default port for OpenVPN is 1194.
  TAB down to **OK** and press **ENTER**

- The next screen will ask you to confirm the Port is set to 1194.
  Make sure **YES** is selected and press **ENTER**

- The next screen will ask you to select a DNS Provider. This can be important if the reason you're looking to have a VPN is for privacy. Many DNS providers log this information and can build a data-set about you. If you don't know which DNS provider to choose simply use Google's DNS provider.
  Use the **DOWN ARROW** to get to Google and press **SPACE** to select. **TAB** to **OK** and press **ENTER**.

- The next screen will allow you to set up your Public IP or DNS entry. If you have a DHCP assigned IP address from your Internet Service Provider, you can use that address, but you'll need to change the client-side configuration file every time your public address changes. There are several ways to setup a Dynamic DNS record using a Dynamic DNS provider.

  Since we are going to be accessing the WAN-side of the pod using a private IP address, we will need to select the default public address that is shown and then we will manually change the actual VPN connection IP address in the configuration file.
  TAB to **OK** and press **ENTER**

- The next screen alerts you that the Server and HMAC keys will be generated.
  Press **ENTER** on **OK**

- Since we will be opening a port on our router to redirect to our Raspberry Pi we can be vulnerable to attacks since we are exposing our device to the internet. What this step will do is enable unattended upgrade of security patches.

  Basically your Raspberry Pi will check daily for new security updates and update itself. You should periodically reboot for some updates to apply.

  Most importantly … Use strong passwords!
  Press **ENTER** on OK

- The next screen whether you wish to enable unattended upgrades of security patches.
  Make sure **YES** is selected and press **ENTER**

- The next screen tells you that the installation is complete and you will need to run the **pivpn add** command to create a "OVPN Profile" for each user to access from their devices (see the next section).
  Press **ENTER** on **OK**

- On the Reboot screen, T**A**B to **YES** and press **ENTER**

- The last screen is the confirmation to reboot.
  Press **ENTER** on **OK** to reboot the Pi and start the OpenVPN services.

Part II - Create your OpenVPN Client File

- Once you have rebooted your Raspberry Pi again, login with the user name pi and the password raspberry

- Then, run the following command:

```
sudo pivpn add
```

This will create a **.ovpn** file which we will need to transfer to the client devices.

This file contains a generated key that is used for logging in to the server.

You can use this file for every device or you can generate new .**ovpn** files by executing the command again with a different client name.

- When asked Enter a Name for the Client: enter **myvpnclient**
- When asked How many days should the certificate last? Press **ENTER** the default which is **1080**
- When asked Enter the password for the client: enter **Password1**
  The password will not show up on the screen
- When asked to Enter the password again to verify: enter **Password1**

This password you just created will need to be entered each time you use your VPN client to connect to your Raspberry Pi VPN server. Use a **strong and long** passphrase since the client .ovpn encryption file and the passphrase are the weaknesses are for someone hacking your Raspberry Pi VPN Server. Of course, keep your configuration/encryption file safe. [For this lab we will be using a simple password that is easy to remember … this is NOT recommended for a production VPN.]

You can see the .ovpn file(s) by typing the command:

```
dir /home/pi/ovpns
```

Part III – Setting up the Router for OpenVPN access

Now the fun part … getting OpenVPN working through your router. There are two steps; the first is to port forward the OpenVPN port to your Raspberry Pi's private IP address, and to setup a static route in order for packets to be redirected back out.

Step 1. Port Forwarding on your Router

Go back to the computer where you configured the router initially and open a browser to the IP address of the router (which should be 192.168.1.1).

You will want to forward your Raspberry Pi's VPN port on your router. The default port you need to forward will be 1194 unless you changed this port in the PiVPN setup.

You'll need to find the "Port Forwarding" screens for you router and add the following:

Application Name:	**`OpenVPN`** (actually it can be any name you wish to use)
Port From / To:	**`1194`** (or whichever port you entered when setting up OpenVPN)
Local IP:	192.168.1.2 << the private IP address that you put the Raspberry Pi on >>
Protocol:	**`UDP`** (if this gives you problems, choose both UTP and TCP

Step 2. Build a Static IP Route Back to the OpenVPN Subnet

You need to forward back packets that came from the OpenVPN subnet (which is 10.8.0.0/24 by default).

You'll need to find the Static IP Route screens for your router and add the following route:

Route Name:	**`OpenVPN-Back`** ← if asked for
Network IP:	**`10.8.0.0`** ← This is assuming you didn't change the default subnet
Netmask:	**`255.255.255.0`**
Gateway:	**`192.168.1.1`** << the private IP address of the router >>
Interface:	**`LAN`** (might be different for each router)

Save the settings and reboot the router.

One note, it can take as long as 5 minutes for the router to establish a connection with the Raspberry Pi OpenVPN service and route packets properly. Be patient.

Step 3. Get the IP Settings for the WAN Side of the Router

You will need to find the WAN address that was assigned to your router from the lab's DHCP server. On a DD-WRT router, for example, the WAN address is listed in the upper right corner of the web page. Every router's firmware might be different. Make a note of this address … this is the WAN address that you will need to enter in the .OVPN file as the "outside" address that the OpenVPN will connect to.

Part IV – Installing an OpenVPN Client and Change the Public IP Address in the .OVPN File

You will need to transfer the .ovpn file created to your client devices which you will be using to connect to your Raspberry Pi VPN server.

- Insert a USB flash drive into a USB port on the Pi
- To find out what the name of the USB drive is, run the following command: **`lsblk`**
- You should see an entry under **NAME** that should be …

  `sda`
  `|___sda1`

  The name on the lower line is the name of the device, in this case it's **`sda1`** … make a note of it
- Make a directory under the **/mnt** folder named "**usb**": **`sudo mkdir /mnt/usb`**
- Mount the USB disk onto the folder you just created: **`sudo mount /dev/sda1 /mnt/usb`**

  Note the folder name after **/dev** is the **NAME** you noted above (**`sda1`** in this example)
- Copy the **.ovpn** file to the USB disk: **`sudo cp /home/pi/ovpns/* /mnt/usb`**
- To see the files on the USB flash drive, type the command: **`dir /mnt/usb`**
- Unmount the USB flash drive: **`sudo umount /mnt/usb`**
- Remove the USB flash drive from the Pi

There are many OpenVPN clients to choose from. We will use the official OpenVPN software that is recommended for Windows computers. On a Mac, the recommended software is Tunnelblick.

Install the Community Edition of the OpenVPN Client

1. Go to the "**Outside**" computer which should be on the lab network (it should be on the same network as the WAN side of the router). Make sure you are logged in on an administrator's account.

2. Ping the **WAN address** of your router to make sure you can see it (you may have to "allow" ping requests on the WAN side of your router).

3. Go to https://openvpn.net/community-downloads/ to download the basic OpenVPN client (there is another client called: OpenVPN Connect which may not work correctly). Click on the appropriate link for your version of Windows (the links will change as the client is updated):

   Windows 7/8/Server 2012 Installer (NSIS) –

   Use **openvpn-install-2.4.9-i601-win7.exe**

   Windows 10/Server 2016/Server 2019 Installer (NSIS) –

   Use **openvpn-install-2.4.9-i601-win10.exe**

4. After downloading the client, run the installer using all of the default settings. You will also be asked to install the TAP adapter.

5. Plug in the flash drive and open the drive to see the files

6. Open another File Manager window and change the folder to:

   C: → Program Files → OpenVPN → Config

   Copy the **.OVPN** file(s) from the flash drive to this folder

7. You will need to edit this file to change the settings for the client with the proper certificates and other configuration options:
   - Right-click on **.ovpn** on the desktop and click on the **Edit With Notepad++** menu option (if you don't have Notepad++ you should install it. You could use Notepad, but it won't display as well).

   - Make the following changes to the file:

     Change **remote xxx.xxx.xxx.xxx 1194**
     The **xxx.xxx.xxx.xxx** is the address that PiVPN picked up during installation
     To **remote *Router's WAN IP address* 1194** ← Use your router's WAN IP address

   - Save the file and exit

   ***Note**: If you get a message saying the file is protected and do you want to open it **ADMINISTRATOR MODE**, click on **YES** then try saving the file again.*

8. Double-click on the **OpenVPN GUI** icon on the desktop which will start the service and place an icon in the notification area

9. Right-click on the icon and click **CONNECT**.

10. A pop-up screen will ask for a password. Use the password you created when you build the .ovpn file, which should be **Password1**.

11. You will see a progress screen show up which should show **COMPETE** at the end and the OpenVPN icon should now be green.

12. Check your IP settings again with **IPCONFIG**
    You should see another Ethernet connection with the IP address on your LAN

13. Ping the inside IP address of your router and the IP address of the computer on the inside of your LAN (make sure the Windows firewall on the computer is turned off).

    To disconnect from OpenVPN, right click on the **OpenVPN GUI** icon in the system tray and click the **Disconnect** menu item.

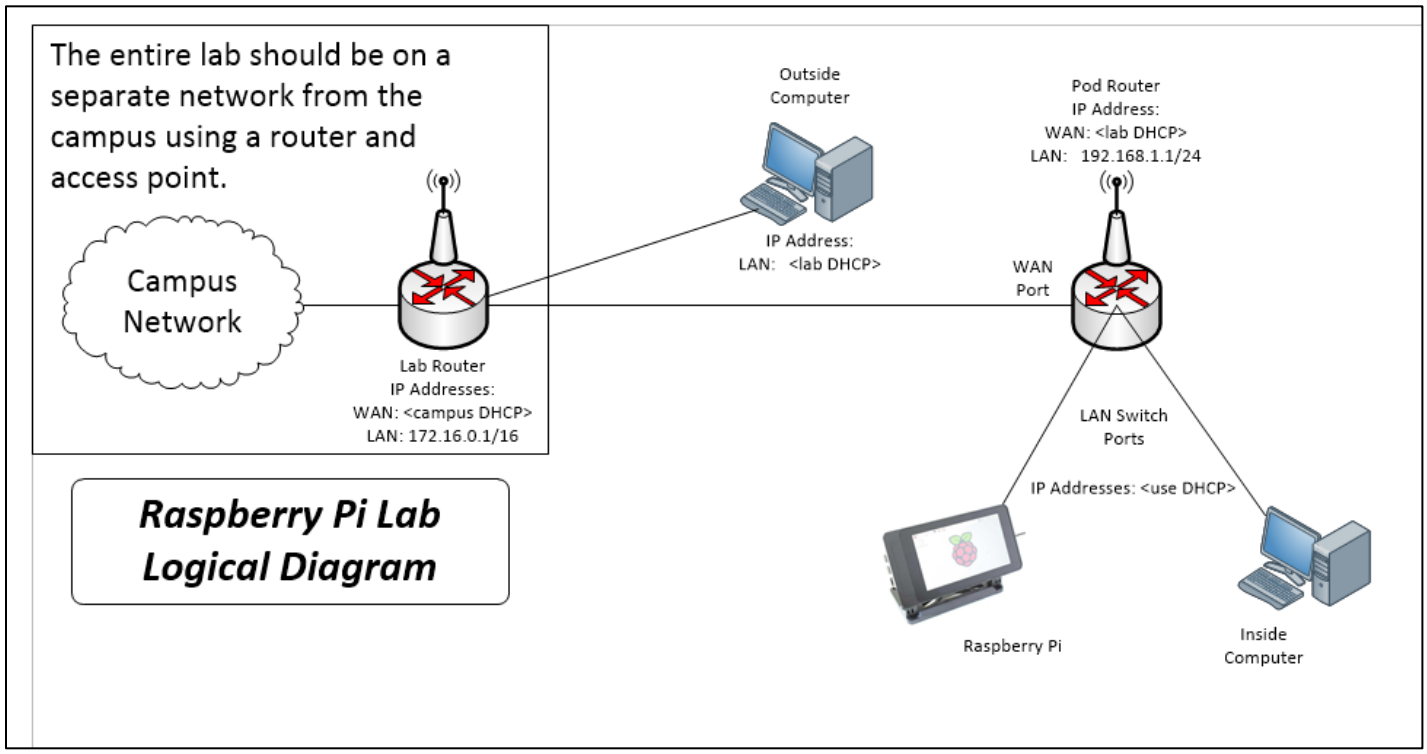
Setting Up FreeRADIUS Using a Raspberry Pi

Environment Setup

You will need the following:

- Raspberry Pi (you can use either a "headless" Pi or one with the 7" built in display)
- SD Card with the current version of Raspbian Lite (the version without the GUI)
- Laptop computer (must have wireless capability) to configure the PI and the router and to connect and test the RADIUS server
- Small Office/Home Office Router/Switch/Access Point (DLink, Linksys, Netgear, etc) to connect the Raspberry Pi and computers to the "outside"
  These instructions assume you'll be using DD-WRT, but the RADIUS setup instructions are basically the same on most of the routers.
- CAT5 cables to run between the Pi, the laptop and the router
- If you are using the Pi with the 7" display, you will also need a keyboard
- Android phone or tablet to connect to RADIUS server (optional)

Below is the logical diagram for the lab pod.



1. Connect the Inside computer to one of the switch ports on the router.

2. Configure the Router/Switch/Access Point with the following IP addresses:
   WAN-Side: Use **DHCP** *which it will get from the lab's DHCP server*
   LAN-Side: Use **192.168.1.1** as the router's IP address
   Make sure the router is set to provide IP addresses using DHCP
   It is important to check the DHCP range that is configured because you will need to manually set a Static IP address on the Raspberry Pi. Make sure the starting address is ***NOT 192.168.1.2*** *it should be something like* ***192.168.1.50*** *or* ***192.168.1.100*** *the number of addresses or the last address is not critical*

3. Image the Raspberry Pi OS (Rasbian) onto the MicroSD card. Use the "Lite" version (no GUI).

4. Connect the Raspberry Pi to one of the switch ports on the router.
   If you have built the Raspberry Pi using the Smart Pi Touch case with the 7 inch touchscreen, then you will need to connect a USB Keyboard. If you are using the Raspberry Pi headless, then you will need to use SSH to connect.

5. Connect the Outside computer to the lab network

6. Power up the Raspberry Pi by plugging it into the power supply.

Part 1 - Installing the PiVPN Implementation of the OpenVPN Protocol

OpenVPN is an open source implementation of an SSL-based VPN. It is a client/server foundation that requires SSL certificates (public/private key infrastructure). Since it is open source, the server and clients can be implemented on many devices, such as routers, servers and virtual appliances. **OpenVPN** Technologies has a simple to implement appliance that can be installed on bare metal or run as a VM. The server software is free, but only allows for two concurrent connections. If you want more than two concurrent connections, you must purchase additional licenses. Third-party implementations do not have this restriction.

PiVPN is a bash-script that takes care of much of the heavy lifting that would be required to install **OpenVPN** on a Raspberry Pi. It makes life pretty easy.

Installation of **PiVPN** (The software we will be using as our VPN server) is a breeze. You simply have to run just one command to install **PiVPN**. This assumes you already have a Raspberry PI OS up and running. You only need the lite version. If you are using the Pi with an attached monitor, keyboard and mouse you could use the full version with the GUI … the setup is the same.

> *If you do not have a monitor and keyboard, then you can use an IP scanner, such as* **IPScan** *to find the IP address from a computer attached to the same network in order to connect to the Pi using SSH.*
>
> *Use a terminal program, such as* **PuTTY** *to* **SSH** *into the Raspberry Pi.*

- Login to the Raspberry Pi, the default user name is **pi** and use **raspberry** as the password.

- Update Raspberry Pi OS (Rasbian) …

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

  … Patience, this can take quite a bit of time.

Set the correct keyboard and location information (the default is a UK keyboard and location)

- At the command prompt type: **`sudo raspi-config`**

- On the following screens, do the following:

    - From the menu options, choose option **4: Localisation Options**
    - From the next menu, choose option **I1: Change Locale**
    - From the list of locales, use the **DOWN-ARROW** key to:
      … deselect: **en_GB.UTF-8 UTF-8** … by pressing the **SPACE** bar
      … choose: **en_US.UTF-8 UTF-8** … by pressing the **SPACE** bar
    - **TAB** down to **OK** and press **ENTER**
    - Use the **DOWN-ARROW** key to select **en_US.UTF-8**
    - **TAB** down to **OK** and press **ENTER**
    - On **DEFAULT-LOCALS** select **NONE**, then **TAB** down to **OK** and press **ENTER**
    - You will be returned to the Configuration Menu

    - From the menu options, choose option **4: Localisation Options**
    - From the next menu, choose option **I3: Change Keyboard Layout**
    - Use the **UP ARROW** key to select: **Generic 104-key PC** and press **ENTER**
    - On the **Keyboard Layout** screen, **DOWN ARROW** to select **OTHER** and press **ENTER**
    - On the **Country of Origin** screen, **DOWN ARROW** to select **English (US)** and press **ENTER**
    - On the **Keyboard Layout** screen, **UP ARROW** to select **English (US)** and press **ENTER**
    - On the next screen, **Keyboard to Function**, make sure **The Default for the Keyboard Layout** is selected and press **ENTER**
    - On the **Compose Key** screen, **,** make sure **No Compose Key** is selected and press **ENTER**
    - Press **TAB** to **FINISH** and press **ENTER**

- You will need to know the IP address of the Raspberry Pi (which is the RADIUS server).
  At the command prompt, type the command:
  `ifconfig eth0` ← and make a note what the IP address is

  Normally you would want to setup a static IP address on the Raspberry Pi running FreeRADIUS. While it's not very difficult to edit the interface configuration file, since this is a simple lab setup, we'll just use the address assigned by DHCP from the router.

Part 1 - Installing the Basic FreeRADIUS Server without MySQL and DaloRADIUS GUI

"*The FreeRADIUS Server is a daemon for unix and unix-like operating systems which allows one to set up a radius protocol server, which can be used for Authentication and Accounting various types of network access. To use the server, you also need a correctly setup client which will talk to it, including terminal servers, Ethernet Switches, Wireless Access Points or a PC with appropriate software.*"

Taken from FreeRADIUS Guide/FAQ at https://wiki.freeradius.org/guide/FAQ#freeradius-overview_what-is-freeradius-and-what-is-it-supposed-to-do

In this first part, we will install and configure the basic FreeRADIUS server. The basic server will work well for small environments where the need for more user security and a captive portal web management console are not necessary … such as a home or small business.

Larger environments or environments where better user management and security (MySQL server) and a more robust configuration engine are needed, you can install MySQL and DaloRADIUS (a web-based GUI front end).

> *A note about security … when a user authenticates with the RADIUS server, the connection between the client and the authentication server is encrypted over a tunnel using EAP/PEAP. What is not secure is the user configuration file on the RADIUS server (the Pi). This file is not encrypted and the user names and passwords are stored in the file in plain text. For a more secure way to manage users and passwords, we would need to use either the MySQL database server or an authentication service such as LDAP.*

1. Install **FreeRADIUS**:
   ```
   sudo apt-get install freeradius
   ```

2. Setup the **Client** configuration file:
   ```
   sudo nano /etc/freeradius/3.0/clients.conf
   ```

   Scroll down about half-way into the document to the **blank space above the line** that says:
   ```
   #IPv6 Client
   ```

   and type in the following statements:
   ```
   client 0.0.0.0/0 {          ← This will allow any client on any network to be the authenticator
      secret    = radpass      ← This is the shared secret password used by the authenticator
      shortname = Router       ← This is a short name representing the authenticator client
   }
   ```
   Save the file by pressing **CTRL+O** and press **ENTER** to accept the existing file name
   Exit the nano editor by pressing **CTRL-X**

Notes:
- *… using 0.0.0.0/0 is NOT a good idea for the authenticator's IP address. you should use the exact IP address of the router, or if there will be several authenticators, then use a subnet address. We'll change the address after installation and testing.*
- *… the secret password should be more complex than what we are using here.*
- *… everything is case sensitive*
- *… do NOT forget to put in the { } where indicated*

3. Setup the **Users** configuration file:

   `sudo nano /etc/freeradius/3.0/users`

   Scroll down to the end of the end of file and type the following statement (case sensitive):

   `raduser Cleartext-Password := "raduserpw"`

   This will start of the list of users/passwords
   You will have an entry for each user who will be authenticated through RADIUS.
   … and since these are in a plain text document, they would be easy to see and steal, which is why you should use either MySQL or LDAP for authentication

   Save the file by pressing **`CTRL+O`** and press **`ENTER`** to accept the existing file name
   Exit the nano editor by pressing **`CTRL-X`**

4. Stop and start the FreeRADIUS service:

   `sudo service freeradius stop`
   `sudo service freeradius start`

5. There is a test built into FreeRADIUS that allows you to make sure that the FreeRADIUS service works correctly.

   `sudo radtest raduser raduserpw 127.0.0.1 0 testing123`

   where:

   `raduser` ← is the user you created in the user's file
   `raduserpw` ← is the password for **raduser**
   `127.0.0.1` ← is the address for LOCALHOST, which tests the network stack
   0 ← is theNAS-Port number (typically 0)
   `testing123` ← the "secret" password that will be exchanged with the router (**`testing123`** is the test password built into the config file)

   The test will return some values … the one that is the most important is the last line:

   `Received Access-Accept Id xx from 127.0.0.1:1812 to 0.0.0.0:0 length 20`

   Make sure you have **`Access-Accept`** … if you do, then FreeRADIUS is working correctly.

6. You should also check the FreeRADIUS log file:

   `sudo tail /var/log/freeradius/radius.log`

   If you see the last line:

   `Info: Ready to process request`

   … everything should be ready to setup the router as the authenticator.

7. Change the Client configuration file to have the actual IP subnet for the authenticators

   Change the **Client** configuration file:

   `sudo nano /etc/freeradius/3.0/clients.conf`

   Scroll down to where you added the client configuration, and make the following change:

   from **`client 0.0.0.0/0 {`**

   to **`client 192.168.1.1/24 {`** ← use the IP address of your router

   Save the file by pressing **`CTRL+O`** and press **`ENTER`** to accept the existing file name
   Exit the nano editor by pressing **`CTRL-X`**

8. Stop and start the FreeRADIUS service:

   `sudo service freeradius stop`
   `sudo service freeradius start`

9. Run the FreeRADIUS test again, only this time, use the settings you have configured

   `sudo radtest raduser raduserpw` ***`RADIUS-IP-address`*** **`0 radpass`**

   where:

`raduser`	← the user you created in the user's file
`raduserpw`	← the password for **raduser**
`RADIUS-IP-address`	← the IP address of the FreeRADIUS server (the Raspberry Pi)
`radpass`	← the "secret" password that will be exchanged with the router

   The test will return some values … the one that is the most important is the last line:
   `Received Access-Accept Id xx from <<IP>>:1812 to 0.0.0.0:0 length 20`

   Make sure you have **`Access-Accept`** … if you do, then you can setup the router to connect.

Part 2 - Setting up the Router to be a FreeRADIUS Authenticator

***Note**: The following section has been written for DD-WRT. However, most routers will have a similar set of configuration screens that should match the values indicated below.*

The material in this section was taken from:
https://www.myopenrouter.com/article/configuring-netgear-wgr614l-dd-wrt-radius-client

Setup the wireless environment from the laptop computer:

1. Open the router's configuration page

2. Go to to the **Wireless** tab and then click on the **Basic Setting** subtab
   Setup the wireless network using your own **SSID** and set the channel to **AUTO** (typical setup)

3. From the **Wireless** tab click on the **Wireless Security** subtab

4. Set the Wireless Security parameters as shown (a screen shot is shown at the bottom of the page)

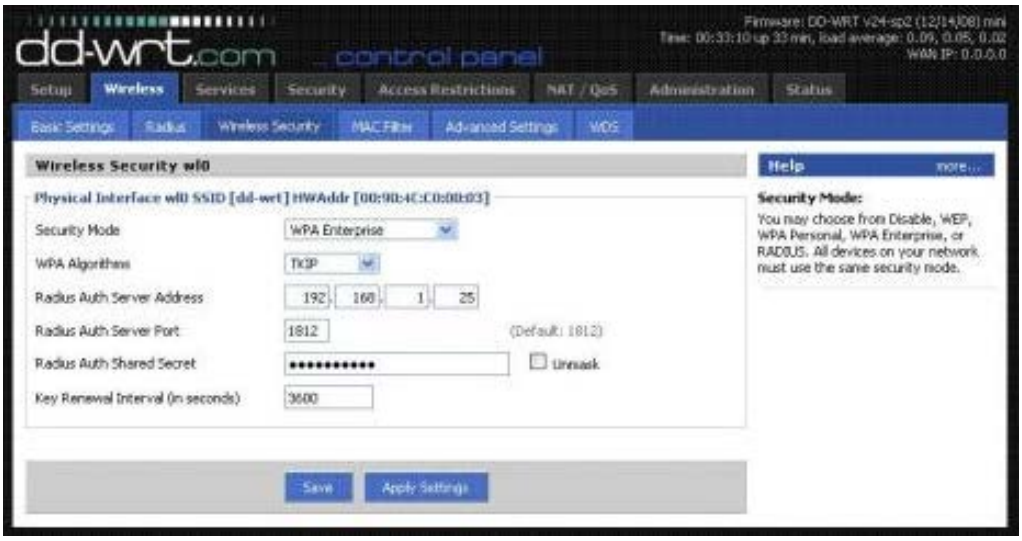
   Security Mode: **WPA2 Enterprise** (you might see WPA2-EAP/WPA-EAP)
   WPA Algorithm: **TKIP+AES** (if you only have AES or TKIP … select TKIP)
   Radius Auth Server Address: ***The IP address of the FreeRADIUS Server*** (the Raspberry Pi)
   Radius Auth Server Port: **1812**
   Radius Auth Shared Secret: **radpass** ← this is the password you entered in step 2
   Key Renewal Interval (in seconds): **3600**

   Save and **Apply** the settings.

5. Reboot the router … **Administration** tab / **Management** subtab → click on the **Reboot Router** button on the bottom of the window

6. Go back to the **FreeRADIUS** server and check the log again:
   ```
   sudo tail /var/log/freeradius/radius.log
   ```

   Make sure you do not have any errors.



Part 3 - Setting up the Wireless Device to Authenticate to FreeRADIUS

Windows computers need to be manually setup to connect to a RADIUS-authenticated wireless.

- If the laptop computer is connected to the wired network, disconnect it from the wired network
- Go to **Control Panel** and click on **Network and Sharing Center**
- Click on **Set up a New Connection or Network**
- Click on **Manually Connect to a Wireless Network**
- Set the following:
  - Network Name: **The SSID of your wireless network**
  - Security Type: **WPA2-Enterprise** ← Use the pull-down list arrow
  - Encryption Type: **AES**
- Click **NEXT**

- Once the settings have been saved, click **Change Connection Settings**
- On the **Wireless Network Properties** page, click on the **Security** tab
- Under Choose a Network Authentication method, there should already be the entry
  Microsoft Protected EAP (PEAP)
- Click on the **Settings** button next to the selection

- On the **PEAP** properties page, uncheck **Validate Server Certificate**
- Under **Select Authentication Method**, there should be the entry
  Secured Password (EAP-MSCHAPV2)
- Click on the **Configure** button next to the selection
- Uncheck the box next to: **Automatically use my Windows logon name and password**
  May already be unchecked
- Click the **OK** button to close the Authentication Properties window
- Click the **OK** button to close the Protected EAP Properties

- Click the **Advanced Settings** button
- On the 802.1x tab, check the **Specify Authentication Mode** checkbox
- Leave the selection **User or Computer Authentication**
- Click the **OK** button to close the Advanced Settings window
- Click the **OK** button to close the Wireless Properties window
- Click the **CLOSE** button on the manual connection window
- Close the **Network and Sharing Center window**

- Click on the wireless icon in the taskbar and select the wireless network you just created
- You will be asked for a user name and password
  - User Name: **raduser**
  - Password: **raduserpw**

You should now be connected

Android Phone:

- Open the settings window (the "gear" at the top right after swiping down from the top)
- Select **WiFi**
- Select your access point
- Set the following:
  - EAP Method: **PEAP** ← Should already be selected
  - Phase 2 Authentication: **MSCHAPV2**
  - CA Certificate: **Do not validate** ← Don't worry about warning message
  - Identity: **raduser** ← The user you created in FreeRADIUS
  - Anonymous Identity: **nobody** ← This masks the encrypted user name / password
  - Password: **raduserpw** ← The password associated with the user
  - Press **Connect**