****WireGuard VPN Setup & Management for Raspberry Pi Routers****

**This document outlines the process of setting up a WireGuard VPN server and configuring Raspberry Pi devices to act as routers connected to the VPN, each with its own subnet.**

****1. Setting Up the Server-Side Script****

**1. Create the script file:**

**vi server-vpn.sh**

**2. Copy and paste the server-side script (server-vpn.sh) into the file.**

**3. Make it executable:**

**chmod +x server-vpn.sh**

**4. \*\*Move to a system path for easy access (This is very essential tomake navigation easier):**

**sudo mv server-vpn.sh /usr/local/bin/server-vpn**

****2. Commands and Usage (Server-Side)****

**These commands are executed on the VPN server.**

****Install & Setup the VPN Server:****

**sudo server-vpn install**

**\* Installs WireGuard and required dependencies.**

**\* Generates server keys.**

**\* Configures the WireGuard server.**

**\* Enables firewall rules and starts the VPN.**

****Add a Raspberry Pi Router Client:****

**sudo server-vpn add raspberrypi\_name**

**\* Generates keys for the Raspberry Pi client.**

**\* Assigns a unique subnet to the Raspberry Pi.**

**\* Configures the WireGuard server to accept the Raspberry Pi.**

**\* Creates a client configuration file for the Raspberry Pi.**

**\* Displays the location of the configuration file.**

****Remove a Raspberry Pi Router Client:****

**sudo server-vpn remove raspberrypi\_name**

**\* Removes the Raspberry Pi client's configuration.**

**\* Removes the client from the WireGuard server.**

**\* Logs the removal time.**

****Monitor Client Connections:****

**sudo server-vpn log**

**\* Checks for active connections every 10 seconds.**

**\* Logs each successful handshake.**

****Enable Auto-Logging on Boot:****

**This ensures the VPN logging service is always running:**

**sudo server-vpn service**

**\* Creates a systemd service to run the logging script at boot.**

****3. Configuration Details****

**Server Configuration (`/etc/wireguard/wg0.conf`):**

**\* Server IP: `10.10.0.1/16`**

**\* Port: `51820`**

**\* Encryption: Uses private/public key pairs**

**\* Public IP Detection: Automatically detected via default network route.**

****Client Configuration Example (Raspberry Pi):****

**This is how a generated client file (`/etc/wireguard/clients/raspberrypi\_name.conf`) will look like this:**

***[Interface]***

***PrivateKey = CLIENT\_PRIVATE\_KEY***

***Address = 10.10.X.1/24 # X will be a unique number for each Pi***

***DNS = 1.1.1.1***

***[Peer]***

***PublicKey = SERVER\_PUBLIC\_KEY***

***Endpoint = SERVER\_IP:51820***

***AllowedIPs = 0.0.0.0/0, ::/0***

***PersistentKeepalive = 25***

**4. Logging & Security Features**

**\* ✅ Logs when a client is added or removed.**

**\* ✅ Logs client connection activity.**

**\* ✅ Prevents duplicate script execution to avoid conflicts.**

**\* ✅ Uses `iptables` for enhanced firewall security.**

**5. Setting Up the Raspberry Pi (Client-Side)**

**These steps are executed on each Raspberry Pi device.**

****1. Prepare the Raspberry Pi:****

**\* Install Raspberry Pi OS on your Raspberry Pi.**

**\* Enable SSH access to your Raspberry Pi.**

****2. Create the client-side script:****

***vi client-vpn.sh***

****3. Copy and paste the client-side script (client-vpn.sh) into the file.****

****4. Make the script executable:****

***chmod +x client-vpn.sh***

****5. Edit the client-vpn.sh script:****

**sudo nano client-vpn.sh**

**Important:Replace the following variables with the correct values for your Raspberry Pi:**

**LOCAL\_NETWORK\_INTERFACE`: The name of the interface connected to your local network (e.g., eth1, wlan0).**

**INTERNET\_INTERFACE: The name of the interface connected to the internet (e.g., eth0, wlan0).**

****6. Transfer the client configuration file:****

**Copy the generated client configuration file (`/etc/wireguard/clients/raspberrypi\_name.conf`) from the server to the Raspberry Pi using `scp` or another file transfer method:**

***scp username@your-server-ip:/etc/wireguard/clients/raspberrypi\_name.conf ./***

**Replace `username` with your server username and `your-server-ip` with your server's public IP.**

**Move the configuration file to the correct location and rename it:**

***sudo mv raspberrypi\_name.conf /etc/wireguard/wg0.conf***

****7. Run the client-side script:****

**sudo ./client-vpn.sh**

**This script will:**

**\* Install WireGuard, `dnsmasq`, and `iptables-persistent`.**

**\* Enable IP forwarding.**

**\* Configure NAT to allow devices on the local network to access the internet through the VPN.**

**\* Configure `dnsmasq` to act as a DHCP server for the local network.**

**\* Start the WireGuard interface.**

****6. Connecting Devices to the Raspberry Pi Router****

**\* Connect your devices to the Raspberry Pi's local network (either via Ethernet or Wi-Fi).**

**\* Configure your devices to obtain an IP address automatically via DHCP.**

**\* Your devices should now be able to access the internet through the VPN and communicate with other devices on the VPN network.**

****7. Troubleshooting****

**Check the VPN Status (Server):**

***sudo systemctl status wg-quick@wg0***

**Restart the VPN (Server):**

***sudo systemctl restart wg-quick@wg0***

**Check VPN Logs (Server):**

***sudo cat /var/log/wireguard\_client.log***

**Check WireGuard Status (Raspberry Pi):**

***sudo wg show wg0***

**Check IP Forwarding (Raspberry Pi):**

***sysctl net.ipv4.ip\_forward***

**Check `iptables` Rules (Raspberry Pi):**

***sudo iptables -L***

***sudo iptables -t nat -L***

**Check Routing Table (Raspberry Pi):**

***route -n***

****8. Verifying the VPN Connection****

**Check your public IP address:**

**\* Before connecting to the VPN, run `curl -s ifconfig.me` on a device connected to the Raspberry Pi's local network.**

**\* After connecting to the VPN, run the same command again.**

**\* The IP address should change to your VPN server's public IP address.**

**Ping devices on other subnets:**

**\* Try pinging devices on the main VPN network (10.10.0.0/16) and on other Raspberry Pi subnets.**

****9. Important Notes****

**\* Interface Names: Double-check and replace the interface names (`eth0`, `eth1`, `wlan0`) in the `client-vpn.sh` script with the correct names for your Raspberry Pi.**

**\* Firewall: You may need to adjust the Raspberry Pi's firewall rules further to allow specific types of traffic between the local network and the VPN.**

**\* Subnet Range: The `get\_next\_subnet` function has a limited range of subnets (10.10.1.0/24 to 10.10.20.0/24). If you need more subnets, you'll need to adjust the `last\_octet` and `max\_octet` variables in the `server-vpn.sh` script.**

**This documentation provides a comprehensive guide to setting up and managing your WireGuard VPN with Raspberry Pi routers. Remember to test thoroughly and adapt the scripts to your specific environment.**

****Updated on: 18th February 2025****