[How-To] Stand Up WireGuard P2P between two sites

Overview of the Setup

- **JAX01 (Your Homelab):**
  - **Network:** Multiple VLANs.
  - **VPN VM:** Will be connected to a dedicated VLAN for VPN traffic.
  - **Local Networks:** Accessible alongside the VPN.
- **JAX02 (Friend's Homelab):**
  - **Network:** Single non-VLAN network (`192.168.60.0/22`).
  - **VPN VM:** Connected to the main network.
  - **Local Networks:** Accessible alongside the VPN.
- **Objective:**
  - **JAX01:** Access both local VLANs and JAX02's network without disrupting local internet traffic.
  - **JAX02:** Access both local network and JAX01's network without disrupting local internet traffic.
  - **Avoid DHCP Conflicts:** Ensure that DHCP servers on both sides do not interfere with each other.

Prerequisites

1. **Ubuntu 24.04 Server VMs:**
   - **SITE01 VM:** Connected to a local network.
   - **SITE02 VM:** Connected to a local network.
2. **Static IP Addresses:**
   - Ensure both VPN VMs have static IPs within their respective networks to maintain consistent VPN connectivity.
3. **Firewall Access:**

- Ensure that UDP port `51820` (default WireGuard port) is open on both networks' firewalls to allow VPN traffic.
4. **Access to Network Equipment:**
   - Ability to create and manage networks at both sites.

Step-by-Step Installation and Configuration

1. Install WireGuard on Both Servers

On Both SITE01 and SITE02 VPN VMs:

1. **Update Package Lists:**
   bash
   `sudo apt update`
2. **Install WireGuard:**
   bash
   `sudo apt install wireguard -y`

2. Generate WireGuard Keys

On Both Servers:

1. **Generate Private and Public Keys:**
   bash
   `umask 077`
   `wg genkey | tee privatekey | wg pubkey > publickey`
   - **Files Created:**
     - `privatekey`: Keep this secure; never share.
     - `publickey`: Share this with your peer.
2. **Exchange Public Keys:**
   - **SITE01's VPN VM Public Key:** Send to SITE02's VPN VM.
   - **SITE02's VPN VM Public Key:** Send to SITE01's VPN VM.

3. Configure WireGuard on Both Servers

SITE01 VPN VM Configuration:

1. **Create WireGuard Configuration File:**
   bash

   `sudo nano /etc/wireguard/wg0.conf`
2. **Add the Following Configuration:**
   ini

   ```
   [Interface]
   PrivateKey = <JAX01_PrivateKey>
   Address = 10.6.0.5/24
   ListenPort = 51820
   SaveConfig = true


   [Peer]
   PublicKey = <JAX02_PublicKey>
   AllowedIPs = 192.168.60.0/22, 10.6.0.10/32
   Endpoint = <JAX02_Public_IP>:51820
   PersistentKeepalive = 25
   ```

   - **Replace:**
     - `<JAX01_PrivateKey>` : Content of `privatekey` on JAX01.
     - `<JAX02_PublicKey>` : Public key from JAX02.
     - `<JAX02_Public_IP>` : Public IP or dynamic DNS of JAX02's network.
     - `10.10.10.2` : IP assigned to JAX02's VPN interface.
3. **Enable IP Forwarding:**
   bash
   `Copy code`
   `sudo sysctl -w net.ipv4.ip_forward=1`
   - To make it persistent:
     bash
     `Copy code`
     `sudo nano /etc/sysctl.conf`
     Add:

     `Copy code`

```
net.ipv4.ip_forward = 1
```

4. **Set Up Firewall Rules (Using UFW as Example):**
   bash

   Copy code

   ```
   sudo ufw allow 51820/udp
   sudo ufw enable
   ```

5. **Start and Enable WireGuard:**
   bash

   Copy code

   ```
   sudo systemctl start wg-quick@wg0
   sudo systemctl enable wg-quick@wg0
   ```

JAX02 VPN VM Configuration:

1. **Create WireGuard Configuration File:**
   bash

   Copy code

   ```
   sudo nano /etc/wireguard/wg0.conf
   ```

2. **Add the Following Configuration:**
   ini

   ```
   [Interface]
   PrivateKey = <JAX02_PrivateKey>
   Address = 10.6.0.10/24
   ListenPort = 51820
   SaveConfig = true


   [Peer]
   PublicKey = <JAX01_PublicKey>
   AllowedIPs = 10.5.40.0/22, 10.6.0.5/32
   Endpoint = <JAX01_Public_IP>:51820
   PersistentKeepalive = 25
   ```

   - **Replace:**
     - `<JAX02_PrivateKey>` : Content of `privatekey` on JAX02.
     - `<JAX01_PublicKey>` : Public key from JAX01.
     - `<JAX01_Public_IP>` : Public IP or dynamic DNS of JAX01's VPN VLAN interface.

3. **Enable IP Forwarding:**
   bash

   Copy code

   ```
   sudo sysctl -w net.ipv4.ip_forward=1
   ```

   - To make it persistent:
     bash
```

```
Copy code
```
```
sudo nano /etc/sysctl.conf
```
Add:

```
Copy code
```
```
net.ipv4.ip_forward = 1
```
4. **Set Up Firewall Rules (Using UFW as Example):**
   bash
   ```
   Copy code
   ```
   ```
   sudo ufw allow 51820/udp
   sudo ufw enable
   ```
5. **Start and Enable WireGuard:**
   bash
   ```
   Copy code
   ```
   ```
   sudo systemctl start wg-quick@wg0
   sudo systemctl enable wg-quick@wg0
   ```

# 4. Configure Routing and Firewall Rules

## On JAX01 (Your Homelab):

1. **Configure Routing to Allow Access to JAX02's Network:**
   - **Assumption:** Local VLANs are managed correctly, and routing is handled by the main router/firewall.
2. **Add Routes for JAX02's Network via VPN:**
   - If using UFW, ensure that forwarding rules allow traffic between VLANs and WireGuard.
   - Example UFW Rules:
     bash
     ```
     Copy code
     ```
     ```
     sudo ufw allow from 10.10.10.0/24 to 192.168.60.0/22
     sudo ufw allow from 192.168.60.0/22 to 10.10.10.0/24
     ```

## On JAX02 (Friend's Homelab):

1. **Configure Routing to Allow Access to JAX01's Network:**
   - Similar to JAX01, ensure that UFW allows traffic between the VPN and the local network.
   - Example UFW Rules:

```bash
```
Copy code

```
sudo ufw allow from 10.10.10.0/24 to 192.168.60.0/22
sudo ufw allow from 192.168.60.0/22 to 10.10.10.0/24
```

5. Prevent DHCP Conflicts

Issue Experienced Previously: DHCP server at JAX02 took over JAX01's network, disrupting home Wi-Fi.

Solution:

- **Isolate VPN Traffic:** Use the dedicated VLAN on JAX01 for VPN traffic, ensuring that DHCP servers on JAX02 do not interfere with JAX01's networks.
- **Ensure No DHCP is Advertised over the VPN:**
  - WireGuard should only handle traffic routing, not DHCP services.
  - Verify that no DHCP server is running on the VPN interfaces.

Verify DHCP Services:

1. **Check for DHCP Servers on JAX01 VPN Interface:**
   bash
   Copy code

   `sudo systemctl status isc-dhcp-server`
   - Ensure that DHCP servers are not binding to the VPN VLAN interface (`eth0.10`).
2. **Check for DHCP Servers on JAX02 VPN Interface:**
   bash
   Copy code

   `sudo systemctl status isc-dhcp-server`
   - Ensure that DHCP servers are not binding to the WireGuard interface (`wg0`).

6. Adjust WireGuard AllowedIPs for Split Tunneling

Objective: Ensure that only traffic meant for the VPN tunnel goes through WireGuard, while local traffic uses the existing network routes.

JAX01 VPN VM Configuration:

- **AllowedIPs for JAX02's Network:**
  ini

  `Copy code`

  `AllowedIPs = 192.168.60.0/22, 10.10.10.2/32`
- **This configuration ensures:**
  - Traffic destined for `192.168.60.0/22` (JAX02's network) goes through the VPN.
  - Traffic destined for the WireGuard interface (`10.10.10.2`) is directly routed.

JAX02 VPN VM Configuration:

- **AllowedIPs for JAX01's Network:**
  ini

  `Copy code`

  `AllowedIPs = 10.10.10.0/24, 192.168.60.0/22`
- **This configuration ensures:**
  - Traffic destined for `10.10.10.0/24` (JAX01's VPN VLAN) goes through the VPN.
  - Traffic destined for `192.168.60.0/22` (local network) uses the local route.

Note: Avoid using `0.0.0.0/0` in `AllowedIPs` to prevent all traffic from routing through the VPN, which could disrupt internet access.

---

7. Start and Enable WireGuard on Both Servers

On Both JAX01 and JAX02:

1. **Enable and Start WireGuard:**
   bash

   `Copy code`

   `sudo systemctl enable [email protected]`
   `sudo systemctl start [email protected]`
2. **Verify WireGuard Status:**
   bash

   `Copy code`

   `sudo wg show`
   - **Expected Output:**
     - Interface details.
     - Peers with correct public keys and allowed IPs.

---

8. Configure Proxmox VMs to Use the VPN

On JAX01:

1. **Assign the VPN VM to the Dedicated VLAN:**
   - In Proxmox, edit the network settings of the VPN VM to connect to the bridge associated with VLAN `10`.
2. **Ensure Proper IP Assignment:**
   - The VPN VM should have an IP like `10.10.10.1/24` on the VLAN interface.

On JAX02:

1. **Connect the VPN VM to the Main Network:**
   - Ensure the VPN VM is connected to the `192.168.60.0/22` network with a static IP (e.g., `192.168.60.10`).

---

9. Testing the VPN Tunnel

From JAX01:

1. **Ping JAX02's VPN IP:**
   bash
   Copy code
   `ping 10.10.10.2`
2. **Access JAX02's Local Network:**
   bash
   Copy code
   `ping 192.168.60.10  # Example device on JAX02`
3. **Access Local VLANs on JAX01:**
   - Ensure that accessing local VLANs on JAX01 works as before.

From JAX02:

1. **Ping JAX01's VPN IP:**
   bash
   Copy code
   `ping 10.10.10.1`
2. **Access JAX01's Local VLANs:**
   bash

```
Copy code
```
```
ping 10.10.10.X  # Replace X with a device on JAX01's VLAN
```

3. **Ensure Local Internet Access:**
   - Browse the internet or ping external sites to confirm traffic is using JAX02's gateway.

---

10. Troubleshooting Tips

1. **Check WireGuard Status:**
   bash
   ```
   Copy code
   ```
   ```
   sudo wg show
   ```
   - Ensure that peers are connected and data is being transferred.
2. **Verify Firewall Rules:**
   - Ensure that UFW or other firewalls are not blocking VPN traffic.
   - Use `sudo ufw status` to review current rules.
3. **Check Routing Tables:**
   bash
   ```
   Copy code
   ```
   ```
   ip route
   ```
   - Ensure that routes for the VPN are correctly set up and do not override default gateways.
4. **Inspect Logs:**
   bash
   ```
   Copy code
   ```
   ```
   sudo journalctl -u [email protected]
   ```
   - Look for any errors or warnings related to WireGuard.
5. **Ensure No IP Overlaps:**
   - Verify that the VPN IP ranges (`10.10.10.0/24`) do not overlap with any existing network ranges.
6. **Restart WireGuard if Necessary:**
   bash
   ```
   Copy code
   ```
   ```
   sudo systemctl restart [email protected]
   ```

---

11. Security Best Practices

1. **Use Strong Keys:**
   - Ensure WireGuard keys are securely generated and stored.
2. **Restrict AllowedIPs:**
   - Only allow necessary IP ranges through the VPN to minimize exposure.

3. **Keep Systems Updated:**
   - Regularly update Ubuntu and WireGuard to patch any security vulnerabilities.
4. **Monitor VPN Traffic:**
   - Use `sudo wg show` and other monitoring tools to keep an eye on VPN connections.
5. **Backup Configurations:**
   - Keep backups of your WireGuard configurations and keys in a secure location.

Conclusion

By following this comprehensive guide, you should be able to establish a secure and efficient WireGuard VPN tunnel between your homelabs (**JAX01** and **JAX02**). This setup ensures that:

- **Local Network Access:** Both locations can access each other's local networks without interference.
- **Internet Traffic:** Internet-bound traffic remains routed through each respective site's gateway, maintaining normal internet functionality.
- **Network Isolation:** The dedicated VLAN on **JAX01** isolates VPN traffic, preventing DHCP conflicts and maintaining network stability.

Remember: Always verify configurations and test thoroughly to ensure network stability and security. If you encounter issues, refer back to the troubleshooting section or consult WireGuard's official documentation for more advanced configurations.

Additional Resources:

- WireGuard Official Documentation

- Proxmox VE Networking

- OPNsense VLAN Setup *(if applicable)*

Feel free to reach out if you need further assistance or encounter specific issues during the setup process!

Revision #2
Created 6 November 2024 18:31:52 by Mike Leffring
Updated 6 November 2024 19:22:43 by Mike Leffring