WireGuard VPN

- Introduction
- Types of installations
- Prerequisites
- Step 1 – Install WireGuard on Ubuntu 22.04
- Step 2 – Configure the WireGuard
- Step 3 – Server Networking and Firewall Configuration
- Step 4 – Linux and macOS Clients Setup
- Step 5 - Windows Clients Setup
- Step 6 - Add the Client Peer to the Server
- Step 7 – Linux and the macOS Clients
- Step 8 – Linux Clients (using .conf file)
- Step 9 – Windows Clients (Using .conf file)

Introduction

Before we begin discussing how to set up a secure WireGuard VPN on Ubuntu 22.04, let's briefly understand – What WireGuard VPN is.

WireGuard is a modern VPN (Virtual Private Network) technology that uses state-of-the-art cryptography compared to other popular VPN solutions, such as IPsec and OpenVPN.

It is faster and easier to configure than its counterparts. WireGuard can be used on Linux, Windows, Android, and macOS.

In this tutorial, you will set up WireGuard VPN on Ubuntu 22.04.

Types of installations

1) Using Automated script(wireguard-install.sh) link:https://github.com/Nyr/wireguard-install

2) Manual setup (following this documentation)

Prerequisites

1) Make sure to have an Ubuntu 22.04 server with root or sudo access.

Step 1 – Install WireGuard on Ubuntu 22.04

1) The WireGuard is available from default Ubuntu repositories. You can install it using the below command:

```
sudo apt update
sudo apt install wireguard
```

It will install the WireGuard module and tools too.

Step 2 – Configure the WireGuard

1) Both **wg** and **wg-quick** command-line tools will allow you to configure. Even manage the WireGuard interfaces. Each device in the **WireGuard** VPN network needs a private as well as a public key. You will run the below command to generate key pair:

It is important that every WireGuard VPN has a public and private key. You can generate the key pair using the below command:

```
wg genkey | sudo tee /etc/wireguard/privatekey | wg pubkey | sudo tee /etc/wireguard/publickey
```

The file's generation will be there in /etc/wireguard directory. You will be able to view the contents of files with cat or less. Make sure to keep the private key safe and secure.

The WireGuard even supports a pre-shared key that adds another layer of symmetric-key cryptography. This key is optional and has to be unique for each pair.

2) After that, configure the tunnel device to route the VPN traffic.

Set-up from the command line using the **ip** and **wg** commands or by creating the configuration file with a text editor.

3) Next, create a new file as `wg0.conf`. Proceed to add the below contents:

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

```
[Interface]
Address = 10.0.0.1/24
SaveConfig = true
ListenPort = 51820
PrivateKey = SERVER_PRIVATE_KEY
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o ens3 -j MASQUERADE
```

The interface can be named anything, it is recommended to use something like **`wg0`** or **`wgvpn0`**. The settings in the interface section have the below meaning:

- Address – With a comma-separated list of v4 or v6 IP addresses. It is for the **`wg0`** interface. Also, use IPs from a range reserved for private networks (10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16).
- The ListenPort - Listening port.
- PrivateKey - Generated by `wg genkey` command.
- SaveConfig - The present state of the interface is saved to the configuration file when shut down if it is set to true.
- PostUp - The command or script executable before bringing the interface up. Like, you are using iptables to enable masquerading. It allows traffic to leave the server, giving the VPN clients access to the Internet.

4) Further, make sure to replace the **`ens3`** after **`-A POSTROUTING`** to match the name of public network interface. You can easily find the interface using the below command:

```
ip -o -4 route show to default | awk '{print $5}'
```

PostDown - The command or script executable before bringing the interface down. Here, iptables rules gets removed, after the interface is down.

5) Normal users should not be able to read both **`wg0.conf`** and **`privatekey`** files. Set the permission to **`600`** using **`chmod`**:

```
sudo chmod 600 /etc/wireguard/{privatekey,wg0.conf}
```

6) After this, bring `wg0` interface up. Do it by attributes specified in the configuration file:

```
sudo wg-quick up wg0
```

The command will show an output similar to:

```
Output

ip link add wg0 type wireguard
wg setconf wg0 /dev/fd/63
ip -4 address add 10.0.0.1/24 dev wg0
ip link set mtu 1420 up dev wg0
iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE
```

7) Continue to check the interface state and configuration using:

```
sudo wg show wg0
```

```
Output

interface: wg0
  public key: r3imyh3MCYggaZACmkx+CxlD6uAmICI8pe/PGq8+qCg=
  private key: (hidden)
  listening port: 51820
```

8) Run **`ip a show wg0`** to verify the interface state:

```
ip a show wg0
```

```
Output

4: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default qlen 1000
    link/none
    inet 10.0.0.1/24 scope global wg0
       valid_lft forever preferred_lft forever
```

The WireGuard can also get managed with `Systemd`.

9) Now, you will bring the WireGuard interface at the boot time by running the below command:

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

Step 3 – Server Networking and Firewall Configuration

1) The IP forwarding should get enabled for NAT to work. Open the `/etc/sysctl.conf` file. Further, add or uncomment the below line:

```
sudo nano /etc/sysctl.conf
```

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

2) Proceed to save the file and apply the changes, by:

```
sudo sysctl -p
```

```
Output

net.ipv4.ip_forward = 1
```

3) Open the UDP traffic on port `51820` if you are using UFW to manage the firewall:

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

Finally, an Ubuntu peer that will act as a server is set up.

Step 4 – Linux and macOS Clients Setup

1) You can check the installation instructions for all supported platforms at https://wireguard.com/install/. Install the package using the distribution package manager and `brew` for macOS.

2) Next, to setup process for a Linux and macOS client is the same as earlier. First, you will generate public and private keys:

```
wg genkey | sudo tee /etc/wireguard/privatekey | wg pubkey | sudo tee /etc/wireguard/publickey
```

3) Then, create the file `wg0.conf` and add below contents:

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

```
Output

[Interface]
PrivateKey = CLIENT_PRIVATE_KEY
Address = 10.0.0.2/24


[Peer]
PublicKey = SERVER_PUBLIC_KEY
Endpoint = SERVER_IP_ADDRESS:51820
AllowedIPs = 0.0.0.0/0
```

If you want to configure the additional clients. You will repeat the same steps. Do it using a different private IP address.

Step 5 - Windows Clients Setup

1) Now, download and install the Windows `msi` package from the WireGuard website.

2) After installation, open the WireGuard application. Then, click on "Add Tunnel" -> "Add empty tunnel..." as shown below:

3) A publickey pair is created automatically and gets displayed on the screen.

4) After that, enter a name for the tunnel and edit the configuration as follows:

```
[Interface]
PrivateKey = CLIENT_PRIVATE_KEY
Address = 10.0.0.2/24

[Peer]
PublicKey = SERVER_PUBLIC_KEY
Endpoint = SERVER_IP_ADDRESS:51820
AllowedIPs = 0.0.0.0/0
```

In this interface section, you will add a new line. It will define the client tunnel Address. Even, in the peer section, add the below fields:

- PublicKey - The public key of Ubuntu server `/etc/wireguard/publickey` file.
- Endpoint - IP address of the Ubuntu server along with a colon and the WireGuard port (51820).
- The AllowedIPs - 0.0.0.0/0

5) After this, click on the Save button.

Step 6 - Add the Client Peer to the Server

1) The Final step is to add the client's public key and IP address to the server. So, run the below command on the Ubuntu server:

```
sudo wg set wg0 peer CLIENT_PUBLIC_KEY allowed-ips 10.0.0.2
```

2) Remember to change `CLIENT_PUBLIC_KEY` with the public key that you generated on client machine `sudo cat /etc/wireguard/publickey`. Further, adjust the client IP address, if different. The Windows users can copy the public key from the WireGuard application.

3) After that, go back to the client machine and bring up the tunneling interface.

Step 7 – Linux and the macOS Clients

1) You will now run the below command to bring up the interface:

```
sudo wg-quick up wg0
```

2) Now you will get connected to the Ubuntu server and traffic from your client machine should get routed from it. You can check the connection using the following command:

```
sudo wg
```

Result:

```
interface: wg0
  public key: gFeK6A16ncnT1FG6fJhOCMPMeY4hZa97cZCNWis7cSo=
  private key: (hidden)
  listening port: 53527
  fwmark: 0xca6c

peer: r3imyh3MCYggaZACmkx+CxlD6uAmICI8pe/PGq8+qCg=
  endpoint: XXX.XXX.XXX.XXX:51820
  allowed ips: 0.0.0.0/0
  latest handshake: 53 seconds ago
  transfer: 3.23 KiB received, 3.50 KiB sent
```

3) Open your browser and type “what is my ip”. You will be able to see your Ubuntu server IP address. Now, to stop tunneling, bring down the wg0 interface:

```
sudo wg-quick down wg0
```

Step 8 – Linux Clients (using .conf file)

1) Download the WireGurd Client config file from the server.

```
apt update & upgrade
apt install wireguard
```

2) Copy the config file into /etc/wireguard

```
cp -p /home/user/vpn/****.conf /etc/wireguard/
```

3) Now, you will bring the WireGuard interface at the boot time by running the below command: Note just add the client file name without .conf

```
sudo systemctl enable wg-quick@****
```

4) Now, you will bring the WireGuard interface up using this command: Note **** is CLIENT FILE NAME

```
wg-quick up ****
```

Result :

```
root@cupcake:/etc/wireguard# wg-quick up cupcake_blue_av
[#] ip link add cupcake_blue_av type wireguard
[#] wg setconf cupcake_blue_av /dev/fd/63
[#] ip -4 address add 10.7.0.2/24 dev cupcake_blue_av
[#] ip link set mtu 1370 up dev cupcake_blue_av
[#] resolvconf -a tun.cupcake_blue_av -m 0 -x
[#] wg set cupcake_blue_av fwmark 51820
[#] ip -6 route add ::/0 dev cupcake_blue_av table 51820
[#] ip -6 rule add not fwmark 51820 table 51820
[#] ip -6 rule add table main suppress_prefixlength 0
[#] ip6tables-restore -n
[#] ip -4 route add 0.0.0.0/0 dev cupcake_blue_av table 51820
[#] ip -4 rule add not fwmark 51820 table 51820
[#] ip -4 rule add table main suppress_prefixlength 0
[#] sysctl -q net.ipv4.conf.all.src_valid_mark=1
[#] iptables-restore -n
```

To disconnect the WireGuard interface:

```
wg-quick down ****
```

Result:

```
root@cupcake:/etc/wireguard# wg-quick down cupcake_blue_av
[#] ip -4 rule delete table 51820
[#] ip -4 rule delete table main suppress_prefixlength 0
[#] ip -6 rule delete table 51820
[#] ip -6 rule delete table main suppress_prefixlength 0
[#] ip link delete dev cupcake_blue_av
[#] resolvconf -d tun.cupcake_blue_av -f
[#] iptables-restore -n
[#] ip6tables-restore -n
```

Step 9 – Windows Clients (Using .conf file)

1) Download the windows installer from the WireGuard page link: https://www.wireguard.com/install/ install WireGuard on a windows PC then import the WireGuard client .config file then press activate the button.

2) When the installation of WireGuard is complete on Windows, click on the “Activate” button. The tunnel status will change to Active once the peers are connected.