# Setting Up WireGuard for Mikrotik Routers Behind NAT

This guide will help you establish a WireGuard VPN that allows your Debian VPS to communicate with your Mikrotik routers, even when they're behind NAT.

## Overview of the Solution

We'll set up a WireGuard server on your Debian VPS with each Mikrotik router connecting as a client. This gives each router a persistent, virtual IP address that's reachable from your VPS regardless of the router's public IP changes.

## Prerequisites

* Debian VPS with public IP address
* Mikrotik routers with RouterOS 7.1 or newer (which includes WireGuard support)
* Administrative access to both the VPS and routers

## Step 1: Set Up WireGuard on the Debian VPS

First, let's install and configure WireGuard on your VPS:

# Update and install WireGuard

sudo apt update

sudo apt install wireguard

# Generate server keys

cd /etc/wireguard

umask 077

wg genkey | tee server\_private.key | wg pubkey > server\_public.key

# Get the keys

SERVER\_PRIVATE=$(cat server\_private.key)

SERVER\_PUBLIC=$(cat server\_public.key)

Now create the server configuration file:

sudo nano /etc/wireguard/wg0.conf

Add the following content:

[Interface]

PrivateKey = <SERVER\_PRIVATE>

Address = 10.8.0.1/24

ListenPort = 51820

SaveConfig = true

# Enable IP forwarding if needed

PostUp = sysctl -w net.ipv4.ip\_forward=1

Replace <SERVER\_PRIVATE> with your actual server private key.

## Step 2: Generate Keys for Each Mikrotik Router

For each router, generate a unique key pair on the VPS:

# For Router 1

wg genkey | tee router1\_private.key | wg pubkey > router1\_public.key

ROUTER1\_PRIVATE=$(cat router1\_private.key)

ROUTER1\_PUBLIC=$(cat router1\_public.key)

# Repeat for each additional router

# Router 2, Router 3, etc.

## Step 3: Add Router Peers to the VPS Configuration

Edit the WireGuard config file again:

sudo nano /etc/wireguard/wg0.conf

Add a [Peer] section for each router:

# Router 1

[Peer]

PublicKey = <ROUTER1\_PUBLIC>

AllowedIPs = 10.8.0.2/32

PersistentKeepalive = 25

# Router 2

[Peer]

PublicKey = <ROUTER2\_PUBLIC>

AllowedIPs = 10.8.0.3/32

PersistentKeepalive = 25

# Add more routers as needed

## Step 4: Start the WireGuard Service on the VPS

sudo systemctl enable wg-quick@wg0

sudo systemctl start wg-quick@wg0

## Step 5: Configure Each Mikrotik Router

For each router, you'll need to configure WireGuard. Here's how to do it via the RouterOS terminal:

# Create the WireGuard interface

/interface wireguard add name=wireguard1 listen-port=13231 private-key="<ROUTER\_PRIVATE\_KEY>"

# Add the VPS as a peer

/interface wireguard peers add interface=wireguard1 public-key="<SERVER\_PUBLIC\_KEY>" allowed-address=10.8.0.0/24 endpoint-address=<VPS\_PUBLIC\_IP> endpoint-port=51820 persistent-keepalive=25s

# Configure IP address (unique for each router)

# Router 1

/ip address add address=10.8.0.2/24 interface=wireguard1

# Router 2

/ip address add address=10.8.0.3/24 interface=wireguard1

# etc.

# Add a route if needed

/ip route add dst-address=10.8.0.0/24 gateway=wireguard1

Alternatively, you can use the Mikrotik WebFig or WinBox GUI to configure these settings.

## Step 6: Update Your PHP API Database

Now you can update your PHP API to use the WireGuard IPs instead of the public IPs:

// Sample code to update your database

$router1\_wg\_ip = "10.8.0.2";

$router2\_wg\_ip = "10.8.0.3";

// etc.

// Update database with these IPs

$query = "UPDATE routers SET ip\_address = ? WHERE router\_id = ?";

$stmt = $pdo->prepare($query);

$stmt->execute([$router1\_wg\_ip, 1]);

// Repeat for other routers

## Step 7: Testing the Connection

From your VPS, test connectivity to each router:

ping 10.8.0.2 # Router 1

ping 10.8.0.3 # Router 2

## Troubleshooting

1. **Connection Issues**: Check firewall rules on both VPS and routers.

# On VPS

sudo iptables -A INPUT -p udp --dport 51820 -j ACCEPT

sudo iptables -A OUTPUT -p udp --sport 51820 -j ACCEPT

1. **NAT Traversal**: If routers are behind strict NAT, try changing the PersistentKeepalive value to a lower number (15-20 seconds).
2. **Mikrotik Interface Status**: Check if the WireGuard interface is running:

/interface wireguard print

1. **VPS WireGuard Status**: Check connection status:

sudo wg show

## Maintaining the System

* If a router's public IP changes, WireGuard will automatically reconnect
* The WireGuard virtual IPs (10.8.0.x) remain constant
* You can modify your PHP API to send commands to these stable WireGuard IPs

By implementing this solution, your Debian VPS will be able to consistently communicate with all your Mikrotik routers regardless of their NAT situation, allowing your PHP API to reliably send commands to each router.