Site-to-Site VPN Implementation

Using pfSense and WireGuard

in VirtualBox Environment

A Comprehensive Implementation Guide

January 11, 2026

Contents

Abstract

This report presents a comprehensive implementation of a site-to-site Virtual Private Network (VPN) using pfSense firewall/router and the WireGuard VPN protocol. The project demonstrates the complete configuration process in a virtualized environment using VirtualBox, covering network architecture design, VPN tunnel establishment, routing configuration, and security considerations. This implementation showcases modern VPN technologies suitable for connecting geographically distributed networks in a secure and efficient manner.

The project addresses key networking concepts including VirtualBox network adapter types, pfSense configuration, WireGuard peer setup, firewall rule management, and routing table configuration. Through detailed step-by-step procedures and troubleshooting methodologies, this report serves as both a technical reference and an educational guide for implementing production-grade site-to-site VPN solutions.

1 Introduction

1.1 Project Overview

In today's interconnected digital landscape, organizations frequently require secure communication channels between geographically separated networks. Site-to-site Virtual Private Networks (VPNs) provide an essential solution for connecting multiple office locations, data centers, or remote facilities through encrypted tunnels over public networks such as the Internet.

This project implements a complete site-to-site VPN solution using pfSense, an open-source firewall and routing platform, combined with WireGuard, a modern and efficient VPN protocol. The entire infrastructure is deployed in a virtualized environment using Oracle VirtualBox, making it an ideal learning platform and proof-of-concept environment.

1.2 Project Objectives

The primary objectives of this implementation are:

- Design and implement a functional site-to-site VPN connecting two separate networks
- Configure pfSense firewalls as VPN endpoints with proper security policies
- Establish encrypted WireGuard tunnels with optimal performance
- Implement proper routing to enable seamless communication between sites
- Configure firewall rules to control and secure inter-site traffic
- Document the complete implementation process with troubleshooting procedures
- Demonstrate network virtualization concepts using VirtualBox

1.3 Use Cases and Applications

Site-to-site VPNs serve numerous practical applications in modern network infrastructure:

- **Multi-site Corporate Networks**: Connecting branch offices to headquarters
- **Disaster Recovery**: Linking primary and backup data centers
- **Cloud Integration**: Connecting on-premises infrastructure to cloud resources
- **Partner Networks**: Secure connections between business partner organizations
- **Development and Testing**: Creating isolated network environments for testing

1.4 Document Structure

This report is organized into comprehensive sections covering theoretical foundations, practical implementation, and operational considerations. Chapter 2 provides essential background on VPN technologies, WireGuard protocol, and pfSense platform. Chapter 3 details the network architecture and design decisions. Chapter 4 covers VirtualBox configuration and network adapter types. Chapters 5-8 present step-by-step implementation procedures for pfSense configuration, WireGuard setup, routing, and firewall rules. Chapter 9 addresses testing and verification, while Chapter 10 provides troubleshooting guidance.

2 Background and Technologies

2.1 Virtual Private Networks (VPNs)

2.1.1 VPN Fundamentals

A Virtual Private Network creates a secure, encrypted connection over a less secure network, typically the Internet. VPNs enable private network communications to traverse public infrastructure while maintaining confidentiality, integrity, and authenticity of data.

The core principles of VPN technology include:

- **Encapsulation**: Original packets are wrapped in additional headers
- **Encryption**: Data is encrypted to prevent unauthorized access
- **Authentication**: Verifying the identity of communicating parties
- **Tunneling**: Creating a virtual point-to-point connection

2.1.2 Site-to-Site vs Remote Access VPNs

VPNs are typically categorized into two main types:

Site-to-Site VPNs (used in this project):

- Connect entire networks to each other
- Always-on, persistent connections
- Transparent to end users
- Router/firewall devices manage the connection
- Suitable for connecting branch offices or data centers

Remote Access VPNs:

- Connect individual users to a network
- On-demand connections initiated by users
- Require client software on user devices
- Used for remote workers and mobile users

2.2 WireGuard VPN Protocol

2.2.1 Introduction to WireGuard

WireGuard is a modern, lightweight VPN protocol designed with simplicity, security, and performance as primary goals. Developed by Jason A. Donenfeld and officially released in 2020, WireGuard has gained rapid adoption due to its superior performance and ease of configuration compared to legacy VPN protocols.

2.2.2 Key Features and Advantages

WireGuard offers several significant advantages:

- **Minimal Codebase**: Approximately 4,000 lines of code compared to hundreds of thousands in IPsec
- **Modern Cryptography**: Uses state-of-the-art cryptographic primitives including Curve25519, ChaCha20, Poly1305
- **High Performance**: Significantly faster than OpenVPN and IPsec due to lightweight implementation
- **Stealth Mode**: Silent protocol that doesn't respond to unauthenticated packets
- **Easy Configuration**: Simple key-based authentication without certificates
- **Roaming Support**: Seamless IP address changes without connection drops
- **Cross-Platform**: Native support in Linux kernel and available on all major platforms

2.2.3 WireGuard Cryptography

WireGuard employs a fixed suite of modern cryptographic protocols:

- **Curve25519**: Elliptic curve Diffie-Hellman key exchange
- **ChaCha20**: Symmetric encryption cipher
- **Poly1305**: Message authentication code
- **BLAKE2s**: Cryptographic hash function
- **SipHash24**: Hash table key function
- **HKDF**: Key derivation function

This fixed cryptographic suite eliminates the complexity and potential vulnerabilities associated with cipher negotiation found in other VPN protocols.

2.2.4 How WireGuard Works

WireGuard operates at the network layer (Layer 3) and creates a virtual network interface. Key operational characteristics:

Key Exchange: Each peer generates a private/public key pair. Peers exchange public keys and configure allowed IP addresses for each peer.

Connection Establishment: WireGuard uses a silent handshake protocol. No connection is established until data is sent, making it stealthy and resistant to probing.

Data Transmission: All data packets are encrypted and authenticated. The protocol includes replay protection and forward secrecy.

Roaming: WireGuard automatically updates peer endpoints as IP addresses change, ideal for mobile devices.

2.3 pfSense Firewall Platform

2.3.1 Overview of pfSense

pfSense is a free, open-source firewall and router platform based on FreeBSD. Originally forked from m0n0wall in 2004, pfSense has evolved into a feature-rich network security solution suitable for everything from home networks to enterprise deployments.

2.3.2 Core Features

pfSense provides comprehensive network security and management capabilities:

- **Firewall**: Stateful packet filtering with flexible rule management
- **Routing**: Static and dynamic routing protocols (BGP, OSPF, RIP)
- **VPN Support**: Multiple VPN protocols including WireGuard, OpenVPN, IPsec
- **Traffic Shaping**: Quality of Service (QoS) and bandwidth management
- **High Availability**: CARP for failover and redundancy
- **Captive Portal**: Guest network authentication and access control
- **Multi-WAN**: Load balancing and failover across multiple internet connections
- **Monitoring**: Real-time traffic graphs and logging
- **Package System**: Extensible architecture with additional packages

2.3.3 pfSense Architecture

pfSense utilizes a modular architecture:

- **Base System**: FreeBSD operating system kernel
- **Web Interface**: PHP-based configuration interface
- **Packet Filter (PF)**: OpenBSD's packet filtering system
- **Package Manager**: System for installing additional features

2.3.4 Why pfSense for Site-to-Site VPN

pfSense is an excellent choice for VPN implementations because:

- Free and open-source with active community support
- User-friendly web interface simplifies configuration
- Native WireGuard support with straightforward setup
- Comprehensive firewall capabilities for security
- Excellent performance even on modest hardware
- Extensive documentation and community resources
- Suitable for both learning and production environments

3 Network Architecture and Design

3.1 Overall Network Topology

The implemented site-to-site VPN connects two geographically separate networks through a secure WireGuard tunnel. Each site consists of a pfSense firewall/router connecting a local area network (LAN) to a wide area network (WAN), with the WAN connections simulated through VirtualBox's NAT Network.

3.2 Site 1 Configuration

3.2.1 Network Addressing

Site 1 utilizes the following IP addressing scheme:

- **LAN Network**: 10.10.11.0/24
- **pfSense LAN Interface**: 10.10.11.1
- **LAN Client VM**: 10.10.11.11 (or DHCP-assigned address)
- **WAN Interface**: 10.0.2.5/24 (NAT Network)
- **WireGuard Tunnel Interface**: 10.69.69.1/24

3.2.2 pfSense Role

The Site 1 pfSense instance functions as:

- Gateway and router for LAN clients
- DHCP server for the LAN network
- Firewall protecting the LAN
- WireGuard VPN server endpoint
- NAT gateway for internet access

3.3 Site 2 Configuration

3.3.1 Network Addressing

Site 2 mirrors Site 1's structure with distinct addressing:

- **LAN Network**: 10.10.12.0/24
- **pfSense LAN Interface**: 10.10.12.1
- **LAN Client VM**: 10.10.12.11 (or DHCP-assigned address)
- **WAN Interface**: 10.0.2.6/24 (NAT Network)
- **WireGuard Tunnel Interface**: 10.69.69.2/24

3.3.2 pfSense Role

The Site 2 pfSense instance provides:

- Gateway and router for LAN clients
- DHCP server for the LAN network
- Firewall protecting the LAN
- WireGuard VPN client endpoint
- NAT gateway for internet access

3.4 WireGuard Tunnel Network

The WireGuard tunnel creates a virtual network segment (10.69.69.0/24) connecting the two sites:

- Dedicated IP space for tunnel endpoints
- Point-to-point Layer 3 connection
- Encrypted communication channel
- Basis for routing between sites

3.5 Traffic Flow Analysis

3.5.1 Inter-site Communication Flow

When a client at Site 1 communicates with a client at Site 2:

1. Client 1 (10.10.11.11) sends packet to Client 2 (10.10.12.11)
2. Packet arrives at Site 1 pfSense LAN interface (10.10.11.1)
3. pfSense routing table directs 10.10.12.0/24 traffic to WireGuard tunnel
4. WireGuard encrypts packet and encapsulates in UDP
5. Encrypted packet sent from WAN interface (10.0.2.5) to destination (10.0.2.6:55555)
6. Site 2 pfSense receives on WAN interface
7. WireGuard decrypts and extracts original packet
8. Site 2 pfSense routes to LAN interface (10.10.12.1)
9. Packet delivered to Client 2 (10.10.12.11)
10. Return traffic follows reverse path

3.5.2 Internet Access Flow

LAN clients access the internet through their respective pfSense gateways:

1. Client sends packet to internet destination
2. pfSense receives on LAN interface
3. Routing table sends to WAN gateway (default route)
4. NAT translates source IP to pfSense WAN IP
5. Packet forwarded to VirtualBox NAT Network gateway
6. Response traffic follows reverse path with NAT translation

3.6 Design Considerations

3.6.1 IP Address Planning

The chosen addressing scheme provides:

- Clear separation between sites (10.10.11.x vs 10.10.12.x)
- Standard /24 subnets allowing 254 hosts per site
- Dedicated tunnel network avoiding conflicts
- RFC 1918 private addressing throughout

3.6.2 Security Architecture

Security is implemented through multiple layers:

- **Perimeter Security**: pfSense firewalls at each site
- **Encryption**: WireGuard tunnel protects inter-site traffic
- **Access Control**: Firewall rules restrict traffic flow
- **Network Segmentation**: Separate LAN, WAN, and tunnel networks

3.6.3 Scalability Considerations

The architecture supports future expansion:

- Additional sites can join the WireGuard mesh
- LAN networks can be subdivided with VLANs
- Multiple subnets can be routed through tunnel
- High availability can be added with CARP

4 VirtualBox Configuration

4.1 VirtualBox Overview

Oracle VM VirtualBox is a powerful, cross-platform virtualization software that enables running multiple operating systems simultaneously on a single physical machine. For this project, VirtualBox provides an isolated, reproducible environment for testing network configurations without requiring physical hardware.

4.2 Virtual Machine Setup

4.2.1 pfSense Virtual Machines

Two pfSense VMs were created with the following specifications:

- **Operating System**: pfSense 2.7.2-RELEASE (FreeBSD amd64)
- **Memory**: 1024 MB RAM minimum (2048 MB recommended)
- **CPU**: 1 virtual CPU (2 recommended for better performance)
- **Storage**: 8 GB virtual hard disk
- **Network Adapters**: 2 adapters (WAN and LAN)

4.2.2 Client Virtual Machines

Two client VMs (one per site) for testing connectivity:

- **Operating System**: Linux (Ubuntu/Debian) or Windows
- **Memory**: 1024 MB RAM
- **CPU**: 1 virtual CPU
- **Storage**: 10 GB virtual hard disk
- **Network Adapter**: 1 adapter (connected to respective LAN)

4.3 Understanding VirtualBox Network Adapters

VirtualBox offers multiple network adapter types, each serving different purposes. Understanding these options is crucial for proper network configuration.

4.3.1 NAT (Network Address Translation)

Characteristics:

- Default adapter type in VirtualBox
- VM can access external networks (internet)
- Host cannot directly access VM

- VMs on separate NAT adapters cannot communicate with each other
- Each VM gets its own private network (10.0.2.0/24 by default)
- VirtualBox acts as router with NAT for each VM independently

Use Cases:

- Single VM needing internet access
- Testing scenarios not requiring inter-VM communication
- Maximum isolation between VMs

Limitations:

- VMs cannot communicate directly with each other
- Complex port forwarding required for inbound connections
- Not suitable for site-to-site VPN testing

4.3.2 NAT Network (Used in This Project)

Characteristics:

- Shared NAT network connecting multiple VMs
- VMs can communicate with each other on same NAT Network
- All VMs share a common network segment
- VirtualBox provides DHCP and routing services
- VMs can access external networks through NAT
- Internal network with external connectivity

Advantages for VPN Project:

- pfSense WAN interfaces can communicate directly
- Simulates real WAN connectivity between sites
- Internet access available for updates and testing
- Simple configuration without manual routing

Configuration:

```
1. VirtualBox > File > Preferences > Network
2. NAT Networks tab
3. Click '+' to add new NAT Network
4. Configure network name and CIDR (10.0.2.0/24)
5. Enable DHCP if desired
6. Apply settings
```

4.3.3 Bridged Adapter

Characteristics:

- VM appears as physical device on host's network
- Receives IP from host network's DHCP server
- Full network access like physical machine
- Visible to other devices on host network

Use Cases:

- Production-like testing environments
- VMs needing to be accessible from physical network
- Testing network services

Considerations:

- Requires permission on host network
- May expose VMs to network threats
- IP addressing managed by external DHCP

4.3.4 Internal Network

Characteristics:

- Isolated network between VMs only
- No external connectivity
- No connection to host machine
- Completely isolated environment

Use Cases:

- Secure, isolated testing
- Malware analysis
- Network segmentation testing

4.3.5 Host-Only Adapter

Characteristics:

- Network between VMs and host only
- No external network access
- VirtualBox provides DHCP server
- Host can communicate with VMs

Use Cases:

- Management access from host
- Testing without internet access
- Secure development environments

4.4 Network Adapter Configuration for This Project

4.4.1 pfSense WAN Adapters (Adapter 1)

Both pfSense VMs' first network adapter configured as:

- **Attached to**: NAT Network
- **Name**: pfSense_NAT (or custom name)
- **Adapter Type**: Intel PRO/1000 MT Desktop (82540EM)
- **Promiscuous Mode**: Deny
- **Cable Connected**: Checked

4.4.2 pfSense LAN Adapters (Adapter 2)

Each pfSense VM's second adapter configured differently:

Site 1 pfSense:

- **Attached to**: Internal Network
- **Name**: LAN1
- **Adapter Type**: Intel PRO/1000 MT Desktop

Site 2 pfSense:

- **Attached to**: Internal Network
- **Name**: LAN2
- **Adapter Type**: Intel PRO/1000 MT Desktop

4.4.3 Client VM Network Adapters

Site 1 Client:

- **Attached to**: Internal Network
- **Name**: LAN1 (same as Site 1 pfSense LAN)

Site 2 Client:

- **Attached to**: Internal Network
- **Name**: LAN2 (same as Site 2 pfSense LAN)

4.5 Why NAT Network Was Chosen

NAT Network was selected for WAN connectivity because:

1. **Inter-VM Communication**: Both pfSense WAN interfaces can reach each other directly, essential for VPN tunnel establishment
2. **Internet Access**: VMs can access external resources for updates and packages
3. **Realistic Simulation**: Mimics real-world WAN connectivity where sites can reach each other over the internet
4. **Simplicity**: No complex host networking configuration required
5. **Isolation**: Separate from host's production network

The key difference between NAT and NAT Network is that standard NAT creates isolated networks per VM (preventing inter-VM communication), while NAT Network creates a shared network allowing VMs to communicate while still providing NAT for external access.

5 pfSense Initial Configuration

5.1 pfSense Installation

5.1.1 Installation Process

The pfSense installation on VirtualBox follows a straightforward process:

1. Download pfSense ISO from official website (pfSense-CE-2.7.2-RELEASE-amd64.iso)
2. Create new VirtualBox VM with FreeBSD (64-bit) as OS type
3. Attach ISO to VM's optical drive
4. Boot VM and follow installation wizard
5. Select default installation options (UFS filesystem)
6. Complete installation and reboot
7. Remove ISO from optical drive

5.1.2 Console Access

After installation, pfSense boots to a console menu providing:

- Interface assignment
- IP address configuration
- DHCP server setup
- Factory reset options
- Shell access
- Reboot and shutdown options

5.2 Interface Assignment

5.2.1 Identifying Network Interfaces

Upon first boot, pfSense detects available network interfaces and prompts for assignment. VirtualBox presents interfaces as:

- em0 (first adapter - WAN)
- em1 (second adapter - LAN)

5.2.2 Assignment Process

From the pfSense console menu:

```
1) Assign Interfaces

Should VLANs be set up now? [y/n]: n

Enter the WAN interface name: em0
Enter the LAN interface name: em1

Do you want to proceed? [y/n]: y
```

5.3 WAN Interface Configuration

5.3.1 Site 1 WAN Configuration

Configure the WAN interface for Site 1 pfSense:

Method 1: DHCP (Recommended for VirtualBox)

```
From console menu:
2) Set interface(s) IP address

Enter the number of the interface: 1 (WAN)
Configure IPv4 address via DHCP? [y/n]: y
Configure IPv6 address via DHCP6? [y/n]: n
Revert to HTTP as the webConfigurator protocol? [y/n]: n
```

The VirtualBox NAT Network DHCP server will assign an IP (typically 10.0.2.5 for first VM).

Method 2: Static IP (Alternative)

```
Configure IPv4 address via DHCP? [y/n]: n
Enter the new WAN IPv4 address: 10.0.2.5
Enter the new WAN IPv4 subnet bit count: 24
Enter the new WAN IPv4 upstream gateway address: 10.0.2.1
Configure IPv6 address via DHCP6? [y/n]: n
Revert to HTTP as the webConfigurator protocol? [y/n]: n
```

5.3.2 Site 2 WAN Configuration

Repeat the same process for Site 2 pfSense. If using DHCP, it will receive 10.0.2.6. If using static, assign 10.0.2.6/24 with gateway 10.0.2.1.

5.4 LAN Interface Configuration

5.4.1 Site 1 LAN Configuration

Configure the LAN interface for Site 1:

```
From console menu:
2) Set interface(s) IP address
```

```

Enter the number of the interface: 2 (LAN)
Configure IPv4 address via DHCP? [y/n]: n
Enter the new LAN IPv4 address: 10.10.11.1
Enter the new LAN IPv4 subnet bit count: 24
For a WAN, press <ENTER> for none: [press ENTER]

Enable DHCP server on LAN? [y/n]: y
Enter the start address of the DHCP range: 10.10.11.100
Enter the end address of the DHCP range: 10.10.11.200
Revert to HTTP as the webConfigurator protocol? [y/n]: n
```

5.4.2 Site 2 LAN Configuration

Configure Site 2 with its distinct addressing:

```
Enter the new LAN IPv4 address: 10.10.12.1
Enter the new LAN IPv4 subnet bit count: 24
Enable DHCP server on LAN? [y/n]: y
Enter the start address of the DHCP range: 10.10.12.100
Enter the end address of the DHCP range: 10.10.12.200
```

5.5 Web Interface Access

5.5.1 Accessing the WebConfigurator

After LAN configuration, the web interface becomes accessible:

- **Site 1**: https://10.10.11.1
- **Site 2**: https://10.10.12.1
- **Default Username**: admin
- **Default Password**: pfsense

Access is available from:

- Client VMs on respective LANs
- Host machine if using Host-Only or Bridged adapter

5.5.2 Initial Setup Wizard

Upon first login, pfSense presents a setup wizard:

1. **Welcome Screen**: Click Next
2. **Netgate Global Support**: Click Next (skip support registration)
3. **General Information**:
    - Hostname: pfsense-site1 (or pfsense-site2)

- Domain: home.arpa
- Primary DNS: 8.8.8.8
- Secondary DNS: 8.8.4.4

4. **Time Server Information**:
   - Timezone: Select appropriate timezone
   - NTP Server: 0.pfsense.pool.ntp.org
5. **Configure WAN Interface**: Verify settings from console configuration
6. **Configure LAN Interface**: Verify settings from console configuration
7. **Set Admin Password**: Change from default 'pfsense' to secure password
8. **Reload Configuration**: Click Reload
9. **Wizard Complete**: Click Finish

5.6 Verifying Basic Connectivity

5.6.1 Testing WAN Connectivity

Verify WAN connectivity from pfSense console:

```
From console menu:
8) Shell

# Test connectivity to NAT Network gateway
ping 10.0.2.1

# Test internet connectivity
ping 8.8.8.8

# Test DNS resolution
ping www.google.com
```

5.6.2 Testing LAN Connectivity

From a client VM on the LAN:

```
# Verify IP address assignment
ip addr show  # Linux
ipconfig      # Windows

# Test gateway connectivity
ping 10.10.11.1  # Site 1
ping 10.10.12.1  # Site 2

# Test internet access through pfSense
ping 8.8.8.8
```

5.6.3 Inter-pfSense WAN Connectivity

Critical for VPN functionality - verify pfSense WAN interfaces can reach each other:

From Site 1 pfSense console:

```
ping 10.0.2.6  # Site 2 WAN IP
```

From Site 2 pfSense console:

```
ping 10.0.2.5  # Site 1 WAN IP
```

If these pings fail, verify NAT Network configuration in VirtualBox. Both pfSense VMs must be on the same NAT Network.

6 WireGuard Tunnel Configuration

6.1 Installing WireGuard Package

6.1.1 Package Installation via Web Interface

WireGuard must be installed as a package on pfSense:

1. Navigate to **System ¿ Package Manager**
2. Click **Available Packages** tab
3. Search for "WireGuard"
4. Click **Install** button for wireguard-kmod package
5. Click **Confirm** to begin installation
6. Wait for installation to complete
7. Package will appear in **Installed Packages** tab

Perform this installation on both Site 1 and Site 2 pfSense instances.

6.1.2 Verifying Installation

After installation, verify WireGuard is available:

- Check that **VPN ¿ WireGuard** appears in navigation menu
- Navigate to **VPN ¿ WireGuard** to access configuration interface

6.2 Creating WireGuard Tunnels

6.2.1 Site 1 Tunnel Configuration

Configure the WireGuard tunnel on Site 1:

1. Navigate to **VPN ¿ WireGuard ¿ Tunnels**
2. Click **Add Tunnel**
3. Configure tunnel parameters:
   - **Enable**: Checked
   - **Description**: wireguard_tunnel
   - **Listen Port**: 55555
   - **Interface Keys**: Click **Generate** to create key pair
   - **Interface Addresses**: 10.69.69.1/24
4. Click **Save**
5. Click **Apply Changes**

Important: Copy and save the public key generated - it will be needed for Site 2 peer configuration.

6.2.2 Site 2 Tunnel Configuration

Configure the WireGuard tunnel on Site 2:

1. Navigate to **VPN ¿ WireGuard ¿ Tunnels**
2. Click **Add Tunnel**
3. Configure tunnel parameters:
   - **Enable**: Checked
   - **Description**: wireguard_tunnel
   - **Listen Port**: 55555
   - **Interface Keys**: Click **Generate** to create key pair
   - **Interface Addresses**: 10.69.69.2/24
4. Click **Save**
5. Click **Apply Changes**

Important: Copy and save this public key as well for Site 1 peer configuration.

6.3 Configuring WireGuard Peers

6.3.1 Understanding Server-Client Model

In this implementation:

- **Site 1 (Server)**: Waits for incoming connections with Dynamic Endpoint enabled
- **Site 2 (Client)**: Initiates connection to Site 1 with static endpoint configuration

This asymmetric configuration is necessary because both sites are behind NAT. One side must act as a server listening for connections.

6.3.2 Site 1 Peer Configuration (Server Side)

Configure the peer representing Site 2:

1. Navigate to **VPN ¿ WireGuard ¿ Tunnels**
2. Expand the tunnel (tun_wg0) section
3. Click **Add Peer**
4. Configure peer settings:
   - **Enable Peer**: Checked
   - **Tunnel**: tun_wg0 (wireguard_tunnel)
   - **Description**: Site2-Peer
   - **Dynamic Endpoint**: **Checked** (Important!)

- **Keep Alive**: 20
- **Public Key**: [Paste Site 2's public key]
- **Allowed IPs**:
  - 10.69.69.2/32 (Site 2 tunnel IP)
  - 10.10.12.0/24 (Site 2 LAN network)

5. Click **Save**

6. Click **Apply Changes**

Key Point: Dynamic Endpoint means Site 1 doesn't need to know Site 2's IP address. It will learn it when Site 2 connects.

6.3.3 Site 2 Peer Configuration (Client Side)

Configure the peer representing Site 1:

1. Navigate to **VPN ¿ WireGuard ¿ Tunnels**

2. Expand the tunnel (tun_wg0) section

3. Click **Add Peer**

4. Configure peer settings:

   - **Enable Peer**: Checked
   - **Tunnel**: tun_wg0 (wireguard_tunnel)
   - **Description**: Site1-Peer
   - **Dynamic Endpoint**: **Unchecked** (Important!)
   - **Endpoint**: 10.0.2.5 (Site 1 WAN IP)
   - **Endpoint Port**: 55555
   - **Keep Alive**: 20
   - **Public Key**: [Paste Site 1's public key]
   - **Allowed IPs**:
     - 10.69.69.1/32 (Site 1 tunnel IP)
     - 10.10.11.0/24 (Site 1 LAN network)

5. Click **Save**

6. Click **Apply Changes**

Critical Configuration: Site 2 must specify Site 1's endpoint (10.0.2.5:55555) since it initiates the connection.

6.3.4 Understanding Allowed IPs

The Allowed IPs configuration is crucial and serves two purposes:
Authentication/Authorization:

- Defines which source IPs are accepted from this peer
- Packets from other IPs are rejected

Routing:

- Defines which destination networks are routed through this peer
- Acts as cryptokey routing table

For site-to-site VPN, Allowed IPs must include:

- Remote tunnel endpoint IP (/32)
- Remote LAN network (/24)

Omitting the remote LAN network will prevent traffic from being routed through the tunnel.

6.4 Interface Assignment

6.4.1 Assigning WireGuard Interface

The WireGuard tunnel creates a virtual interface (tun_wg0) that must be assigned:

1. Navigate to **Interfaces ¿ Assignments**
2. In **Available network ports** dropdown, select **tun_wg0**
3. Click **Add**
4. New interface appears as OPT1 (or next available)
5. Click on the interface name (OPT1)
6. Configure interface:
   - **Enable**: Checked
   - **Description**: WireGuard (or VPN)
   - **IPv4 Configuration Type**: None (already configured in tunnel settings)
   - **IPv6 Configuration Type**: None
7. Click **Save**
8. Click **Apply Changes**

Perform this on both Site 1 and Site 2 pfSense instances.

6.5 Gateway Configuration

6.5.1 Creating Gateway for WireGuard Interface

Configure a gateway for routing through the tunnel:

On Site 1:

1. Navigate to **System ¿ Routing ¿ Gateways**
2. Click **Add**
3. Configure gateway:
   - **Interface**: WireGuard (OPT1)
   - **Address Family**: IPv4
   - **Name**: OPT1GW
   - **Gateway**: 10.69.69.2 (Site 2 tunnel IP)
   - **Monitor IP**: 10.69.69.2
   - **Description**: WireGuard Gateway to Site 2
4. Click **Save**
5. Click **Apply Changes**

On Site 2:

1. Navigate to **System ¿ Routing ¿ Gateways**
2. Click **Add**
3. Configure gateway:
   - **Interface**: WireGuard (OPT1)
   - **Address Family**: IPv4
   - **Name**: OPT1GW
   - **Gateway**: 10.69.69.1 (Site 1 tunnel IP)
   - **Monitor IP**: 10.69.69.1
   - **Description**: WireGuard Gateway to Site 1
4. Click **Save**
5. Click **Apply Changes**

7 Routing Configuration

7.1 Understanding Routing Requirements

For proper site-to-site communication, each pfSense must know how to reach the remote LAN network. This is accomplished through static routes pointing to the WireGuard gateway.

7.2 Static Route Configuration

7.2.1 Site 1 Static Route

Configure routing to Site 2's LAN:

1. Navigate to **System ¿ Routing ¿ Static Routes**
2. Click **Add**
3. Configure route:
   - **Destination network**: 10.10.12.0/24
   - **Gateway**: OPT1GW (10.69.69.2)
   - **Description**: Route to Site 2 LAN via WireGuard
4. Click **Save**
5. Click **Apply Changes**

7.2.2 Site 2 Static Route

Configure routing to Site 1's LAN:

1. Navigate to **System ¿ Routing ¿ Static Routes**
2. Click **Add**
3. Configure route:
   - **Destination network**: 10.10.11.0/24
   - **Gateway**: OPT1GW (10.69.69.1)
   - **Description**: Route to Site 1 LAN via WireGuard
4. Click **Save**
5. Click **Apply Changes**

7.3 Verifying Routing Tables

7.3.1 Checking Routes via Console

Verify routes are properly installed in the routing table:

On Site 1 pfSense:

```
# From shell (option 8)
netstat -rn | grep 10.10.12

# Expected output:
10.10.12.0/24       10.69.69.2          UGS        tun_wg0
```

On Site 2 pfSense:

```
netstat -rn | grep 10.10.11

# Expected output:
10.10.11.0/24       10.69.69.1          UGS        tun_wg0
```

7.3.2 Route Flags Explanation

Understanding routing table flags:

- **U**: Route is up
- **G**: Route is to a gateway
- **S**: Static route (manually configured)

8 Firewall Rules Configuration

8.1 Firewall Rule Fundamentals

8.1.1 pfSense Firewall Philosophy

pfSense implements stateful packet filtering with a default-deny policy:

- Traffic is blocked unless explicitly allowed
- Rules are processed top-to-bottom
- First matching rule wins
- States are maintained for allowed connections

8.1.2 Interface-Based Rules

Firewall rules are applied per interface based on packet direction:

- Rules filter traffic **entering** the interface
- No filtering on outbound (leaving) traffic
- Requires rules on multiple interfaces for bidirectional communication

8.2 WAN Interface Rules

8.2.1 Allowing WireGuard Traffic

Critical rule for establishing VPN connection - Site 1 must accept incoming WireGuard packets:

On Site 1 WAN interface:

1. Navigate to **Firewall ¿ Rules ¿ WAN**
2. Click **Add** (up arrow to add to top)
3. Configure rule:
   - **Action**: Pass
   - **Interface**: WAN
   - **Address Family**: IPv4
   - **Protocol**: UDP
   - **Source**: Any
   - **Destination**: WAN address
   - **Destination Port**: 55555
   - **Description**: Allow WireGuard from Site 2
4. Click **Save**
5. Click **Apply Changes**

Important: Site 2 does not need a WAN rule for WireGuard because it initiates the connection (outbound traffic is not filtered).

8.3 WireGuard Interface Rules

8.3.1 Allowing Tunnel Traffic

Both pfSense instances require rules on the WireGuard interface:

On both Site 1 and Site 2:

1. Navigate to **Firewall ¿ Rules ¿ WireGuard (OPT1)**
2. Click **Add**
3. Configure rule:
   - **Action**: Pass
   - **Interface**: WireGuard
   - **Address Family**: IPv4
   - **Protocol**: Any
   - **Source**: Any
   - **Destination**: Any
   - **Description**: Allow all traffic through VPN tunnel
4. Click **Save**
5. Click **Apply Changes**

8.4 LAN Interface Rules

pfSense typically includes a default LAN rule allowing outbound traffic. This rule allows:

- LAN clients to access the internet
- LAN clients to communicate through VPN tunnel
- LAN clients to access pfSense services

9 Testing and Verification

9.1 Verifying WireGuard Tunnel Status

9.1.1 Checking Tunnel Status

On both pfSense instances:

1. Navigate to **VPN ¿ WireGuard ¿ Status**
2. Verify the following information:
   - **Latest Handshake**: Should show recent timestamp (within last few minutes)
   - **Transfer RX/TX**: Should show data transferred (in KB or MB)
   - **Endpoint**: Should show peer's IP address and port
   - **Allowed IPs**: Should list configured networks

A recent handshake time indicates the tunnel is active and healthy.

9.2 Testing Tunnel Connectivity

9.2.1 Ping Tunnel Endpoints

Test connectivity between tunnel endpoints:

From Site 1 pfSense console:

```
# Option 8 - Shell
ping 10.69.69.2

# Expected result: Successful ping replies
```

From Site 2 pfSense console:

```
ping 10.69.69.1

# Expected result: Successful ping replies
```

If tunnel endpoint pings succeed, the WireGuard tunnel is functioning properly.

9.3 Testing LAN-to-LAN Connectivity

9.3.1 Ping Between pfSense LAN Interfaces

Test routing through the tunnel:

From Site 1 pfSense:

```
ping 10.10.12.1

# Expected result: Successful pings to Site 2 LAN interface
```

From Site 2 pfSense:

```
ping 10.10.11.1

# Expected result: Successful pings to Site 1 LAN interface
```

9.3.2 Testing from LAN Clients

Test end-to-end connectivity between client VMs:

From Site 1 Client VM (10.10.11.11):

```
# Ping Site 2 pfSense LAN interface
ping 10.10.12.1

# Ping Site 2 client VM
ping 10.10.12.11

# Traceroute to see path
traceroute 10.10.12.11
```

Expected traceroute output:

```
traceroute to 10.10.12.11 (10.10.12.11), 30 hops max
1  10.10.11.1 (10.10.11.1)  1.234 ms  # Site 1 pfSense
2  10.10.12.11 (10.10.12.11)  5.678 ms  # Site 2 client
```

9.4 Performance Testing

9.4.1 Bandwidth Testing with iperf3

Test tunnel throughput using iperf3:

On Site 2 Client (server mode):

```
iperf3 -s
```

On Site 1 Client (client mode):

```
iperf3 -c 10.10.12.11 -t 30
```

This tests TCP throughput through the VPN tunnel for 30 seconds.

9.4.2 Latency Testing

Measure round-trip time:

```
ping -c 100 10.10.12.11

# Analyze statistics:
# - min/avg/max latency
# - packet loss percentage
```

9.5 Monitoring and Logging

9.5.1 Real-time Traffic Monitoring

Monitor VPN traffic:

1. Navigate to **Diagnostics ¿ Traffic Graph**
2. Select **WireGuard (OPT1)** interface
3. Observe traffic patterns during testing

9.5.2 Firewall Logs

Review firewall activity:

1. Navigate to **Status ¿ System Logs ¿ Firewall**
2. Filter by interface (WireGuard, WAN, LAN)
3. Look for blocked or passed traffic
4. Identify any connection issues

10 Troubleshooting

10.1 Common Issues and Solutions

10.1.1 Tunnel Won't Establish

Symptom: No handshake shown in WireGuard Status, tunnel endpoints cannot ping each other.

Possible Causes and Solutions:

1. **WAN connectivity issue**:
   - Verify pfSense WAN interfaces can ping each other
   - Ensure both pfSense VMs are on same NAT Network in VirtualBox
   - Check: `ping 10.0.2.5` and `ping 10.0.2.6`
2. **Firewall blocking WireGuard**:
   - Verify WAN rule on Site 1 allows UDP port 55555
   - Check firewall logs for blocked packets
   - Ensure rule is above any deny rules
3. **Incorrect peer configuration**:
   - Verify public keys are correctly exchanged
   - Ensure Site 2 has correct endpoint (10.0.2.5:55555)
   - Check that Site 1 has Dynamic Endpoint enabled
   - Verify Site 2 has Dynamic Endpoint disabled
4. **WireGuard service not running**:
   - Check Status ¿ Services
   - Restart WireGuard service if needed
   - Review system logs for errors

10.1.2 Tunnel Established But No LAN-to-LAN Communication

Symptom: Tunnel IPs (10.69.69.x) are reachable, but LAN networks cannot communicate.

Possible Causes and Solutions:

1. **Missing or incorrect Allowed IPs**:
   - Verify Allowed IPs include remote LAN networks
   - Site 1 peer should have: 10.69.69.2/32, 10.10.12.0/24
   - Site 2 peer should have: 10.69.69.1/32, 10.10.11.0/24
2. **Static routes not configured**:

- Check System ¿ Routing ¿ Static Routes
- Verify routes exist for remote LAN networks
- Test: `netstat -rn | grep 10.10`

3. **Firewall rules blocking tunnel traffic**:
   - Check Firewall ¿ Rules ¿ WireGuard interface
   - Ensure rule allows traffic from/to LAN networks
   - Review firewall logs for blocked packets

4. **Gateway misconfiguration**:
   - Verify WireGuard gateways point to correct tunnel IPs
   - Site 1 gateway should be 10.69.69.2
   - Site 2 gateway should be 10.69.69.1

10.1.3 Intermittent Connection Drops

Symptom: Tunnel works but periodically loses connectivity.

Possible Causes and Solutions:

1. **NAT timeout**:
   - Ensure Keep Alive is set (20 seconds recommended)
   - Keep Alive prevents NAT timeout by sending periodic packets

2. **Network instability**:
   - Check VirtualBox host system resources
   - Verify no network adapter errors in VM logs

10.2 Diagnostic Commands

10.2.1 WireGuard Status Check

```
# From pfSense shell
wg show

# Shows detailed WireGuard interface information:
# - Interface name and public key
# - Peer configuration
# - Latest handshake time
# - Transfer statistics
# - Endpoint information
```

10.2.2 Routing Table Inspection

```
# Display full routing table
netstat -rn

# Filter for specific networks
netstat -rn | grep 10.10.11
netstat -rn | grep 10.10.12
netstat -rn | grep 10.69.69

# Check default gateway
netstat -rn | grep default
```

10.2.3 Firewall State Table

```
# View active connections
pfctl -s states | grep 10.69.69

# View firewall rules
pfctl -s rules
```

10.2.4 Packet Capture

Use pfSense's built-in packet capture:

1. Navigate to **Diagnostics ¿ Packet Capture**
2. Select interface (WAN, LAN, or WireGuard)
3. Set filters (host, port, protocol)
4. Start capture and generate traffic
5. Analyze captured packets

10.3 Troubleshooting Workflow

10.3.1 Systematic Approach

Follow this step-by-step troubleshooting process:

1. **Verify Physical/Virtual Connectivity**:
   - Test WAN interface pings between pfSense instances
   - Confirm VirtualBox network adapter configuration
2. **Check WireGuard Status**:
   - Review VPN ¿ WireGuard ¿ Status
   - Look for recent handshake

- Verify transfer statistics

3. **Verify Routing Configuration**:
   - Check static routes are configured correctly
   - Verify gateways point to correct IPs
   - Test with traceroute/ping

4. **Examine Firewall Rules**:
   - Review rules on all interfaces
   - Check firewall logs for blocks
   - Test with temporary allow-all rules if needed

5. **Review System Logs**:
   - Check System Logs ¿ General
   - Look for WireGuard or network errors
   - Review service status

6. **Use Packet Capture**:
   - Capture traffic on relevant interfaces Identify where packets are dropped or malformed

11 Conclusion

11.1 Project Summary

This project successfully demonstrated the implementation of a site-to-site VPN using pfSense and WireGuard in a VirtualBox environment. The configuration achieved secure, encrypted communication between two geographically separate networks with minimal complexity and high performance.

Key accomplishments include:

- Designing and implementing a complete network topology with two separate sites
- Configuring pfSense firewalls as secure VPN gateways
- Establishing encrypted WireGuard tunnels with proper peer authentication
- Implementing static routing for seamless inter-site communication
- Configuring firewall rules to control traffic flow while maintaining security
- Documenting comprehensive testing and verification procedures
- Creating systematic troubleshooting methodology for common issues

11.2 Technical Achievements

The implementation showcased several important technical concepts:

- **VirtualBox Networking**: Utilized NAT Network and Internal Network adapters to simulate realistic WAN and LAN environments
- **pfSense Configuration**: Demonstrated interface assignment, firewall rules, static routing, and gateway configuration
- **WireGuard Protocol**: Implemented modern VPN protocol with key-based authentication and cryptokey routing
- **Network Security**: Established multiple security layers including encryption, firewall rules, and access controls
- **Troubleshooting Methodology**: Developed systematic approach to diagnosing and resolving network connectivity issues

11.3 Lessons Learned

Throughout the implementation process, several valuable insights emerged:

- **WireGuard Simplicity**: WireGuard's straightforward configuration significantly reduces complexity compared to traditional VPN protocols
- **Importance of Documentation**: Detailed documentation is crucial for understanding complex network configurations and facilitating troubleshooting

- **Virtualization Benefits**: VirtualBox provides an excellent platform for network experimentation without requiring physical hardware
- **Security Layers**: Effective network security requires multiple layers of defense working together
- **Testing Methodology**: Systematic testing at each stage prevents cascading configuration errors

11.4 Future Enhancements

The current implementation can be extended with several enhancements:

- **Additional Sites**: Expand to multi-site mesh topology connecting three or more locations
- **High Availability**: Implement CARP for redundant pfSense gateways
- **Network Segmentation**: Use VLANs to segment traffic within each site
- **Advanced Routing**: Implement dynamic routing protocols (OSPF, BGP) for larger networks
- **Performance Monitoring**: Add detailed monitoring with tools like Prometheus/-Grafana
- **Backup Systems**: Implement automated configuration backups and disaster recovery procedures

11.5 Final Thoughts

This project demonstrates that modern VPN technologies like WireGuard, when combined with powerful platforms like pfSense, enable organizations to build secure, high-performance network infrastructure without excessive complexity. The virtualized testing environment provides an ideal platform for learning, experimentation, and development before deploying to production environments.

The techniques and methodologies documented in this report provide a solid foundation for implementing production-grade site-to-site VPN solutions in real-world scenarios, from small business branch connections to enterprise network infrastructure.

References

1. pfSense Documentation. (2023). *pfSense Firewall Software.* Retrieved from https://docs.netgate.c

2. Donenfeld, J. A. (2023). *WireGuard Protocol Specification.* Retrieved from https://www.wireguard

3. Oracle Corporation. (2023). *VirtualBox User Manual.* Retrieved from https://www.virtualbox.org

4. Free Documentation License. (n.d.). *pfSense Documentation License.* Retrieved from https://docs.netgate.com/pfsense/en/latest/license.html

5. Network Working Group. (2015). *RFC 768: User Datagram Protocol.* IETF.

6. Network Working Group. (1996). *RFC 1918: Address Allocation for Private Internets.* IETF.