

Install WireGuard VPN Server on OpenBSD 7.0

Learn how to install and configure a WireGuard VPN server on OpenBSD 7.0 with step-by-step instructions for secure, fast, and reliable network connections.

Contents

01	Introduction	3
02	Configure WireGuard Server	3
03	Optional: Configure WireGuard Client	7
04	Test your WireGuard VPN Server	9
05	Conclusion	10

Introduction

WireGuard VPN is a free, open-source virtual private network (VPN) solution that implements modern cryptography to secure network connections. It is lightweight, fast, and easy to deploy since it uses public-key exchanges to create a VPN connection.

In this article, you install WireGuard VPN Server on OpenBSD 7.0 and configure a client connection to the server. Private addresses, `10.0.0.1/32`, `10.0.0.2/32` are used as WireGuard server and client addresses, respectively. You can formulate a custom IP class for the VPN network.

Prerequisites

- [Deploy an OpenBSD Server](#)
- SSH and Login as root

Install WireGuard VPN Server.

```
# pkg_add wireguard-tools
```

Install Nano.

```
# pkg_add nano
```

Configure WireGuard Server

First, allow forwarding on your server interfaces with the following commands:

```
# sysctl net.inet.ip.forwarding=1  
# sysctl net.inet6.ip6.forwarding=1
```

Output:

```
net.inet.ip.forwarding: 0 -> 1
net.inet6.ip6.forwarding: 0 -> 1
```

Add the entries to `/etc/sysctl.conf`.

```
# echo "net.inet.ip.forwarding=1" >> /etc/sysctl.conf
# echo "net.inet6.ip6.forwarding=1" >> /etc/sysctl.conf
```

Next, create the WireGuard configuration files directory.

```
# mkdir -p /etc/wireguard
```

Generate Keys

Change to the WireGuard configuration files directory, and set up the `wg0.conf` file that contains all server configurations.

```
# cd /etc/wireguard
```

Generate a new private key.

```
# wg genkey > private.key
```

Then, generate a new public key.

```
# wg pubkey <private.key> public.key
```

Now, create the configuration file.

```
# touch /etc/wireguard/wg0.conf
```

View, and copy the private and public keys.

```
# cat private.key public.key
```

Output: (Your values should be different)

```
EJH+xdUPnD8Wid0kV3YGcEa2kzjqgGo9n7rWmDsUimA=  
1hqTUawZrVDGRu1hwTVKEFz7Ra00Eh+9IWLC3+NXeFU=
```

Now, open and edit the configuration file using a text editor of your choice.

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

Paste the following contents. Replace `Server Private Key` with the private key generated earlier. As well, enter your client IP (local for the VPN network) in the `Allowed IPS =` section, or simply enter `0.0.0.0/0, ::/0` to allow connections from all addresses.

```
[Interface]  
PrivateKey = SERVER-PRIVATE-KEY  
ListenPort = 51820  
  
#Client configuration  
  
[Peer]  
PublicKey = CLIENT-PUBLIC-KEY  
  
#Enter assigned Client local VPN address  
AllowedIPs = 10.0.0.2/32  
  
#Keep the connection alive  
PersistentKeepalive = 25
```

Save and close the file.

Next, open and edit the firewall configuration file at `/etc/pf.conf`.

```
# nano /etc/pf.conf
```

Paste the following contents to allow connections and NAT traffic from the WireGuard interface.

```
pass in on wg0
pass in inet proto udp from any to any port 51820
pass out on egress inet from (wg0:network) nat-to (vio0:0)
```

Save and close the file.

Restart the firewall.

```
# pfctl -f /etc/pf.conf
```

Additionally, create a new hostname file for the WireGuard interface.

```
# nano /etc/hostname.wg0
```

Paste the following contents:

```
inet 10.0.0.1 255.255.255.0 NONE
up
!/usr/local/bin/wg setconf wg0 /etc/wireguard/wg0.conf
```

Now, activate the WireGuard interface.

```
# sh /etc/netstart wg0
```

Run `ifconfig` to confirm if a new `wg0` interface is created.

```
# ifconfig wg0
```

Your output should be similar to:

```
wg0: flags=80c3<UP,BROADCAST,RUNNING,NOARP,MULTICAST> mtu 1420
index 5 priority 0 llprio 3
wgport 51820
wgpkey PJFy3cY+LJf/YQeQ7wvLw+fSDzGy6Qu8kkLJNbrVhUk=
wgpeer YHYGb4q1AyRAINXqzndyqMW5pfqAMvJ/8nwU1S08dQk=
wgpka 25 (sec)
tx: 2072, rx: 0
wgaip 10.0.0.2/32
```

```
groups: wg
inet 10.0.0.1 netmask 0xffffffff broadcast 10.0.0.255
```

Use `wg` to view the current WireGuard server status.

```
# wg
```

Your output should be similar to:

```
interface: wg0
  public key: PJFy3cY+LJf/YQeQ7wvLw+fSDzGy6Qu8kkLJNbrVhUk=
  private key: (hidden)
  listening port: 51820

peer: YHYGb4q1AyRAINXqzndyqMW5pfqAMvJ/8nwJLS08dQk=
  allowed ips: 192.168.2.2/24
  transfer: 0 B received, 2.89 KiB sent
  persistent keepalive: every 25 seconds
```

Optional: Configure WireGuard Client

Depending on your setup, you can configure the WireGuard client on another OpenBSD server or your local machine running Windows, macOS, Linux. For purposes of this article, set up WireGuard client on another OpenBSD and test the connection.

Repeat all the above WireGuard server configuration steps and only change values in the `wg0.conf` file. Also, assign a different local IP Address `10.0.0.2/32` in your `hostname.wg0` file.

Generate client keys.

```
# wg genkey > private.key

#wg pubkey <private.key> public.key
```

Edit the WireGuard configuration file.

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

Paste the following contents:

```
[Interface]
PrivateKey = CLIENT-Private-Key

[Peer]
PublicKey = Server-Public-Key
Endpoint = Server-IP:51820 #Your Vultr Server Public IP Address
AllowedIPs = 0.0.0.0/0, ::/0
```

Enter the generated client private key, then enter the server public key copied earlier. In the `EndPoint =` section, enter your OpenBSD public IP Address.

Edit the `hostname.wg0` file.

```
#nano /etc/hostname.wg0
```

Paste the following code:

```
inet 10.0.0.2 255.255.255.0 NONE
up

!/usr/local/bin/wg setconf wg0 /etc/wireguard/wg0.conf
```

Save and close the file.

Activate the Interface.

```
# sh /etc/netstart wg0
```

Edit your firewall configuration file.

```
# nano /etc/pf.conf
```

Paste the following rules:

```
pass out on egress inet from (wg0:network) nat-to (vio0:0)
```

Save and close the file.

Restart Firewall.

```
# pfctl -f /etc/pf.conf
```

To add new clients on the server, simply repeat the above processes, and include a new `[peer]` section for every client.

Test your WireGuard VPN Server

Your WireGuard VPN Server is up and forwarding packets on your OpenBSD server. Depending on your client computer (whether another server or local computer), test your VPN by pinging your server.

```
# ping Your-WireGuard-Server-Public-IP
```

On the server machine, run `wg` to view the connection details.

```
# wg
```

Your output should be similar to:

```
interface: wg0
  public key: PJFy3cY+LJf/YQeQ7wvLw+fSDzGy6Qu8kkLJNbrVhUk=
  private key: (hidden)
  listening port: 51820

peer: 1tav0yogeBSLQ0D/Y8XhJXxUy6aYzqgNbM+avuH59nE=
  endpoint: 41.75.188.192:46467
  allowed ips: 10.0.0.2/32
  latest handshake: 1 minute, 44 seconds ago
  transfer: 11.68 MiB received, 68.65 MiB sent
  persistent keepalive: every 25 seconds
```

Traffic from the client machine is indicated as `transfer: received`.

Conclusion

You have successfully set up WireGuard VPN server on OpenBSD, all connected clients will have Internet access through the server. Note that the 10.0.0.0/32 addresses are not provided by Vultr, but rather created for the VPN network. To learn more about WireGuard VPN, consider viewing the [official documentation](#).



VULTR

