

# IPTables: Remote-Access VPN (Host-to-Site) - Lab Guide

This lab was developed by Sparta Global for Cybersecurity courses. It was built based on the original lab that was developed for the Labtainer framework by the Naval Postgraduate School, Center for Cybersecurity and Cyber Operations under National Science Foundation Award No. 1438893. This work is in the public domain, and cannot be copyrighted.

## 1 Overview

A VPN is virtual in that it carries information within a private network, but that information is actually transported over a public network.

A VPN is private in that the traffic is encrypted to keep the data confidential while it is transported across the public network.

OpenVPN is an open source tool that can be used to create VPN connections. It uses a custom protocol based on SSL and TLS.

Host-to-Site VPN is similar to host-to-host except that instead of establishing the VPN connection to the server/host you want to communicate with, you will establish the VPN connection with a VPN gateway.

In Host-to-Site, you are able to communicate with the whole network (based on your authorisation) instead of communicating with only one host.

A VPN-Gateway can be a router, firewall or a dedicated server.

## 2 Lab Environment

This lab runs in the Labtainer framework, available at <http://my.nps.edu/web/c3o/labtainers>. That site includes links to a pre-built virtual machine that has Labtainers installed, however Labtainers can be run on any Linux host that supports Docker containers.

From your labtainer-student ( /labtainer/labtainer-student) directory start the lab using:

```
labtainer sparta-vpn2
```

A link to this lab guide will be displayed.

## 3 Network Configuration

IP addresses and routing are configured on all devices.

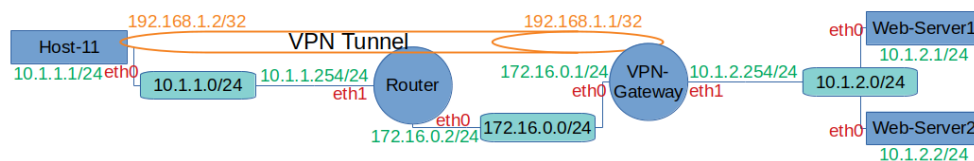


Figure 1: Network topology for routing-basics lab

The network is composed of (find the main components of this network based on its topology):

- 
- 
- 
- 
- 

The OpenVPN application is pre-installed on the host and the server, and the OpenVPN configuration files already exist.

## 4 Credentials

- **Host-11:**
  - **Username:** user-11
  - **Password:** user-11
- **Web-Server1:**
  - **Username:** web-admin
  - **Password:** web-admin
- **Web-Server2:**
  - **Username:** web-admin
  - **Password:** web-admin
- **Router:**
  - **Username:** admin
  - **Password:** admin
- **VPN-Gateway:**
  - **Username:** admin
  - **Password:** admin

## 5 Lab Tasks

### 5.1 Checking/Testing the Initial Configuration

Lets check what we can/can't do in this network.

- On Host-11 (Check IP configuration and network interfaces)

```
ip addr
```

What is the result ? How many interfaces on the this machine ?

- On Router (Check IP configuration and network interfaces)

```
ip addr
```

What is the result ? How many interfaces on the this machine ?

- On VPN-Gateway (Check IP configuration and network interfaces)

```
ip addr
```

What is the result ? How many interfaces on the this machine ?

- On Host-11 (Ping Host-11 -> Web-Server1)

```
ping 10.1.2.1
```

What is the result ? Why ? Is the server (private) network reachable from the host network ?

- On Host-11 (Ping Host-11 -> Web-Server2)

```
ping 10.1.2.2
```

What is the result ? Why ? Is the server (private) network reachable from the host network ?

## 5.2 Capturing the Traffic

Lets use **tcpdump** which is a command line tool that can capture TCP/IP and other packets being transmitted or received over a network interface.

In the Router terminal, run the following command:

```
sudo tcpdump -n -XX -i eth1
```

Lets run tcpdump on Web-Server1 so we can monitor the traffic received on the server:

```
sudo tcpdump -n -XX -i eth0
```

## 5.3 Configuring the VPN - VPN-Gateway Side

- Check VPN configuration

```
ls
```

Notice that there are two files:

- gateway.conf: it contains the OpenVPN configuration that will be used during establishing the VPN connection with the client.

```
cat gateway.conf
```

The file content is as follows:

- \* The first line 'dev tun' is the defining the type of the device/interface that will be created during the VPN establishment. In this case, we are using tunnel interface.
  - \* The second line 'ifconfig 192.168.1.1 192.168.1.2' defines the IP addresses of both end of tunnel.
  - \* The third line 'secret static.key' points to the file where the shared static key is stored.
  - \* The fourth line 'push "route 10.1.2.0 255.255.255.0"' advertises to the vpn-client that this network can be reached through the VPN-tunnel.
- static.key: it contains the same secret key that exists on the client and will be used to establish the VPN connection with the client.

```
cat static.conf
```

- Start the VPN connection on the VPN-Gateway.

```
sudo openvpn --config gateway.conf --daemon
```

## 5.4 Configuring the VPN - Host/Client Side

- Check VPN configuration

```
ls
```

Notice that there are two files:

- client.conf: it contains the OpenVPN configuration that will be used during establishing the VPN connection with the VPN-Gateway.

```
cat client.conf
```

The file content is as follows:

- \* The first line 'remote 10.1.2.1' is the OpenVPN Gateway that it will connect to.
  - \* The second line 'dev tun' is the defining the type of the device/interface that will be created during the VPN establishment. In this case, we are using tunnel interface.
  - \* The third line 'ifconfig 192.168.1.1 192.168.1.2' defines the IP addresses of both end of tunnel.
  - \* The fourth line 'secret static.key' points to the file where the shared static key is stored.
  - \* The fifth line 'route 10.1.2.0 255.255.255.0 192.168.1.1 1' adds a new routing record to route thr traffic to this network through the tunnel interface. The '1' at the end of the line gives this route a higher priority than all other previous routes to the same network.
- static.key: it contains the secret key that will be used to establish the VPN connection with the VPN-Gateway. This is a shared key, that means the VPN-Gateway should have the exact key in order to successfully establish the connection.

```
cat static.conf
```

- Establishing the VPN connection.

```
sudo openvpn --config client.conf --daemon
```

## 6 Testing the connectivity

- On Host-11 (Check IP configuration and network interfaces)

```
ip addr
```

What is the result ? How many interfaces on the this machine ?

- On VPN-Gateway (Check IP configuration and network interfaces)

```
ip addr
```

What is the result ? How many interfaces on the this machine ?

- On Host-11 (Ping Host-11 -> Web-Server)

```
ping 192.168.1.1
wget 10.1.2.1
ping 10.1.2.2
wget 10.1.2.2
```

What is the result ? Why do we used the private IP addresses not the tunnel ones ?

- Check the router terminal to see whether you still can find unencrypted text in the packets captured by the router.  
Check whether you can see the IP addresses (10.1.1.1, 10.1.2.1 or 10.1.2.2). Why can't you see any of them ?
- Check the server terminal and check which IP addresses you can see in the log. Can you see the IP address 10.1.1.1 ? Why ?

## 7 Submission

After finishing the lab, go to the terminal on your Linux system that was used to start the lab and type:

```
stoplab sparta-vpn2
```

When you stop the lab, the system will display a path to the zipped lab results on your Linux system. Provide that file to your instructor, e.g., via the Sakai site.