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

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 conections. It uses a custom protocol based on SSL and TLS.

A remote-access VPN is dynamically created to establish a secure connection between a client and a VPN terminating device. For example, a remote access SSL VPN might be used when you check your banking information online.

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-vpn
```

A link to this lab guide will be displayed.

3 Network Configuration

IP addresses and routing are configured on all devices.

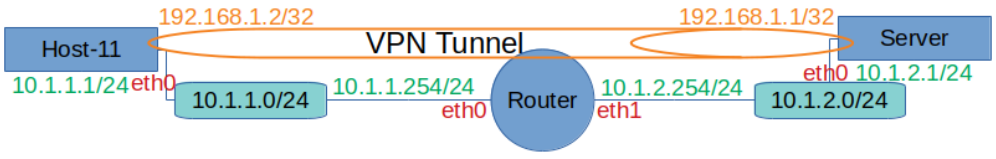

Figure 1: Network topology for routing-basics lab

The network is composed of:

- Host-11: it is the client in this design and it has an IP address 10.1.1.1/24.
- Server: an OpenVPN server and Web-Server at the same time. It has an IP address: 10.1.2.1/24
- Router: It connects the two network without having any direct role in the VPN connection.

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
- **Server**:
  - **Username:** web-admin
  - **Password:** web-admin
- **Router**:
  - **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 Server (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 Host-11 (Ping Host-11 -> Web-Server)

  ```
  ping 10.1.2.1
  ```

  What is the result ?

- On Host-11 (Send HTTP request Host-11 -> Web-Server to get a web page)

  ```
  wget http://10.1.2.1/index.html
  ```

  What is the result ?

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 eth0
```

eth0 is the interface connected to our gateway. You can make sure of that by running **ifconfig** or **ip add**

5.3 Configuring the VPN - Server Side

- Check VPN configuration

  ```
  ls
  ```

  Notice that there are two files:

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

    ```
    cat server.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.

  - 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 server.

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

5.4 Configuring the VPN - Host/Client Side

- Check VPN configuration

  ```
  ls
  ```

  Notice that there are two files:

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

    ```
    cat client.conf
    ```

    The file content is as follows:

    * The first line remote 10.1.2.1 is the OpenVPN server 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.
  - static.key: it contains the secret key that will be used to establish the VPN connection with the server. This is a shared key, that means the server 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 Server (Check IP configuration and network interfaces)

  ```
  ip addr
  ```

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

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

  ```
  ping 192.168.1.2
  ```

  What is the result ? Which device is sending the response in this case ?

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

  ```
  ping 192.168.1.1
  ```

  What is the result ? Which device is sending the response in this case ?

- On Host-11 (Send HTTP request Host-12 -> Web-Server to get a web page)

  ```
  wget http://192.168.1.1
  ```

  What is the result ? Which device is sending the response in this case ?

  Check the router terminal to see whether you still can find unencrypted text in the packets captured by the router.

  Try again the same command but with the main IP address (without VPN) and check the traffic captured on the router terminal.

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-vpn
```

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.