Virtual Private Network with Open Source and Vendor Based Systems

Veeramuthu Rajaravivarma
SUNY, Farmingdale State College, Farmingdale
Rajarav@farmingdale.edu

Abstract

Heavy dependency on vendor-based software is now gradually beginning to shift towards open source software. Many businesses and government agencies are cautiously switching from a vendor-based environment to an open source environment. This paper will focus practical aspects of both Vendor Based VPN Technology and Open Source VPN Technology. This is done by setting up a laboratory using minimal hardware and available vendor based and/or open-source software required to demonstrate Virtual Private Network (VPN) communications in action. As a result, the readers will learn how to set up a vendor based and Open source VPN laboratory and also how to test the connections stability and security in a wired and wireless environment as a home and mobile user.

Vendor Based VPN Technology Laboratory

A Virtual Private Network (VPN) is a secure private network connection that typically uses a public or shared network as its transport. Of course, the most widely known (and common) public network is the Internet. In essence, a VPN connection is a secure "tunnel" between two devices. There are two main components to a VPN connection: the concentrator and the client.

1. The concentrator is typically located in the central hub site of a company, and its function is to terminate the VPN tunnels that are generated from remote devices. The concentrator has at least one interface that is reachable over the Internet.
2. The client is the initiator of the VPN tunnel and is typically located at a remote location. The client can be either software or hardware-based. In either case, the client contacts the concentrator (using the publicly accessible Internet interface) to initiate the VPN tunnel. The two parties then negotiate connectivity settings and a VPN tunnel is established.

When connecting a remote office using VPN, a VPN client creates a secure tunnel to a central VPN concentrator. However, the VPN client is hardware-based and is not specific to any individual user on the network. Instead, the VPN hardware client creates the tunnel and shares that VPN tunnel with multiple users that connect to it on its private LAN interface. The lab exercises accompanying this document are an illustration of site-to-site VPN using the Cisco 3000-series VPN hardware. Other hardware, such as routers from Cisco or other hardware vendors, can also be used to establish site-to-site VPN connectivity. Based on Figure 1, a VPN test laboratory procedure along with configuring

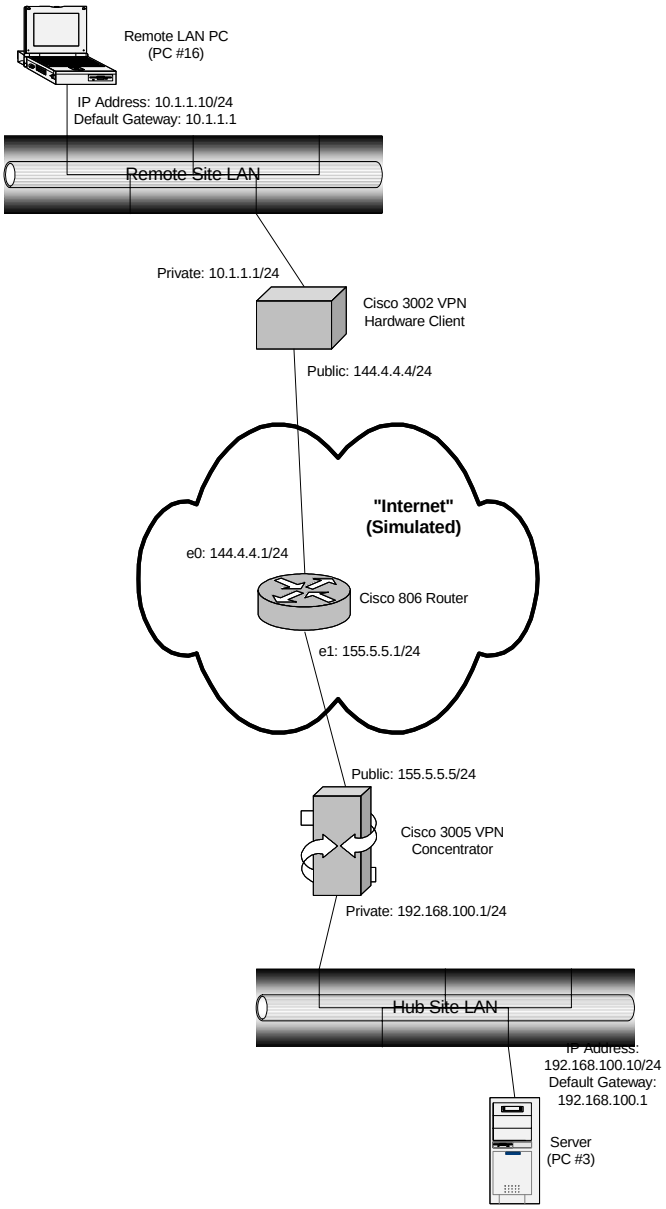

Figure 1. Vendor Based VPN Technology Laboratory

the VPN Hardware Client and Configuring the VPN Concentrator are explained in this section and in [1]. Once implemented, this lab can be used for many different purposes and can be very valuable as a troubleshooting and learning aid. For example, once installed, the reader can disconnect and reconnect the VPN equipment and notice what happens to the environment and what logs the equipment keeps of such occurrences.

To setup this lab, the following equipments are needed (see Figure 1):

- Cisco 3002 VPN Hardware Client
- Cisco 3005 VPN Concentrator
- A router (in this case, a Cisco 806)
- Two PC's (one acting as the client, and one acting as the server) with Windows installed
- 4 CAT-5 RJ-45 patch cables

To set up the VPN tunnel the following steps must executed in order:

- ***Configure the PC's with the following network parameters:***
  1. PC #16 (Remote LAN PC)
     - IP address: 10.1.1.10
     - Subnet mask: 255.255.255.0
     - Default gateway: 10.1.1.1
  2. PC #3 (Server PC)
     - IP address: 192.168.100.10
     - Subnet mask: 255.255.255.0
     - Default gateway: 192.168.100.1

- ***Configure the router to simulate an Internet connection***. This must be configured with the following parameters:
  1. Interface #1 (in this case "e0")
     - IP address: 144.4.4.1
     - Subnet mask: 255.255.255.0
  2. Interface #2 (in this case "e1")
     - IP address: 155.5.5.1
     - Subnet mask: 255.255.255.0

- ***Configure the Cisco 3002 VPN Hardware Client***
  1. Connect PC to console port of the 3002.
  2. Use HyperTerminal to setup a console session with the 3002 – use normal "Cisco" settings.
  3. Login: admin, admin
  4. Navigate thru the menus to configure an IP address for the private interface:
     - *1) Configuration*
     - *2) Interface Configuration*
     - *1) Configure the Private Interface*
     - *1) Interface Setting*
     - *2) Enable using Static IP Addressing*
     - *then, enter the assigned IP address and subnet mask for the private interface (in this case: 10.1.1.1 with a subnet mask of 255.255.255.0)*
     - *Select "back" on all menus until you get to the main menu; then select "exit"*
  5. Connect PC to an Ethernet port on "private" side
     PC will be assigned a DHCP address
  6. Use a web browser to go to the private interface IP address assigned above
     Login: admin, admin
  7. Click on "Click here for main menu"
  8. When logged into the web interface, use the menu on the left side to navigate to the following locations:
     - Configuration
       i. Interfaces
          - Public Interface

a. Select “Static IP” and enter the necessary values (in this case, 144.4.4.4 with a subnet mask of 255.255.255.0)
b. Hit Apply at the bottom of the page

ii. System
- Tunneling Protocols
  a. IPSec
    i. Enter remote server address: (in this case it is 155.5.5.5)
    ii. Enter Group and User usernames and passwords (pre-shared keys)
      1. Group username: “CCSULAB”
      2. Group password: “lab4vpn”
      3. User username: “lab-test3002”
      4. User password: “3002access”
    iii. Hit Apply at the bottom of the page
- IP Routing
  a. Default Gateway
    i. Enter public default gateway address (in this case it is 144.4.4.1)
    ii. Enter Metric = 1
    iii. Hit Apply at the bottom of the page
    iv.
- Management Protocols
- Policy Management
  a. Traffic Management
    i. PAT
      1. Enable
        a. Uncheck “PAT Enabled”
        b. Hit Apply at the bottom of the page

- Administration

i. Access Rights
- Administrators
  a. Three user accounts are preset and cannot be changed
  b. For Admin account: change the password to “cisco”

9. Click on Configuration at left pane
   - Notice the upper right of the right frame says “Save needed”
   - Click on the disk next to “Save needed” to save the current configuration to NVRAM

- ***Configure the Cisco 3005 VPN Concentrator***
  1. Connect via a console cable, similar to how you configured the 3002 Hardware Client.
  2. Enter initial configuration information (use Cisco 3005 users manual for specific configuration instructions):
     - Public IP address: 155.5.5.5
     - Public subnet mask: 255.255.255.0
     - Public default gateway: 155.5.5.1

- Private IP address: 192.168.100.1
- Private subnet mask: 255.255.255.0

3. Using a web browser on a PC on the local subnet of the 3005, enter URL of 3005 Concentrator (using the private interface addresses)
   - http://192.168.100.1
4. Login into 3005 Concentrator
   - Login: admin
   - Password: admin
5. Add VPN group to 3005:
   - On "FAR LEFT" of web page, select:
     i. Configuration
     ii. User Management
     iii. Groups
   - Click on "Add Group"
   - Enter "Group Name" (Same as "IPSec Group Name" as configured in 3002 – in this case it is "CCSULAB")
   - Enter Password (Same as "IPSec Group Password" as configured in 3002 – in this case it is "lab4vpn") and type again to verify it.
   - Select "Internal" from drop-down box
   - Click on "Add"
6. Add VPN Site (User) to 3005:
   - On "FAR LEFT" of web page, select:
     i. Configuration
     ii. User Management
     iii. Users
   - Click on "ADD"
   - Enter "USER NAME" (Same as "IPSec User Name" as configured in 3002 – in this case it is "lab-test3002")
   - Enter Password (Same as "IPSec User Password" as configured in 3002 – in this case it is "3002access")
   - Select "Group" from "Drop-Down" box (Same as "IPSec Group Name" as configured in 3002 – "CCSULAB")
   - Click on "ADD"
   - Click on "Save Needed" on "FAR Right" of web page.
   - Click on "OK".

- When all boxes are configured***, connect Ethernet patch cables (see Figure 1)***
  1. PC #3 (server) to Cisco 3005 "private" interface
  2. 3005 public interface to "e1" interface of Cisco router
  3. 3002 public interface to "e0" interface of Cisco router
  4. PC #16 (client) to "private" interface of Cisco 3002

- When all devices, including the PC's, are configured and connected properly, the ***VPN tunnel should automatically be established*** between the 3002 and the 3005. This can be verified by:

1. Logging into the 3005, then selecting “Administration” then “Administer Sessions”. The VPN tunnel should show up in the list of sessions.
2. Ping from the client PC (#16) to the server PC (#3). To do this, open a DOS command prompt on the client PC, then type: “ping 192.168.100.10”)

Now you should have a working VPN tunnel. There are many features and tricks that can be done with the Cisco products, as well as other vendors’ products

Open Source VPN Technology

OpenVPN is open source software available from Openvpn.net. It is available for most operating systems including Linux, a number of the BSD UNIX’s, Windows XP, Windows 2000, Solaris, and MAC OS X. The software creates a virtual network adapter, which takes packets from the adapter, encrypts them, encapsulates them onto a UDP connection and sends them out to the destination. At the receiving end these packets are unencrypted, authenticated and de-encapsulated. The applications are not aware that a VPN exists and are not able to tell them apart from a WAN consisting of dedicated circuits.

Beyond this OpenVPN uses two basic configurations classified as “Routing” and “Bridging”. Bridging bridges the virtual network device with the actual physical network device in software. In short it makes a machine appear as if it was part of the remote subnet – one broadcast domain. There is only one interface that is created by the OpenVPN installation. It shows up under Windows XP as a win32-tap interface. What determines if the configuration will be bridging or routing depends on how the configuration files for OpenVPN are setup as well as whether interface bridging has been done as described above. The device created for bridging is called a TAP device, which is a virtual Ethernet adapter. Some of the advantages of bridging are broadcasts can travel between remote clients and the LAN. This allows NetBIOS file sharing and browsing in Windows. Furthermore there are no route statements to configure and any protocol that can work with Ethernet can be used including IPv6. The disadvantages are that it does not scale well and it is less efficient than routing [2].

Routing on the other hand requires no bridging of interfaces. Rather it requires the use of an additional private subnet. Routes are required on both the client machine and remote gateway so that packets can be easily moved back and forth between the tun or tap devices. Routing can use either device, which needs to be configured in the client and server configuration files. For routing either tun or tap settings must be used in both the client and server machine configuration files[2]. Using routing is more efficient than using bridging and it scales well.

Open Source VPN Technology Laboratory Setup

The software required to setup a VPN using OpenVPN can be downloaded from the OpenVPN (http://openvpn.net/) web site for a number of different operating systems. For the classroom VPN experience [3], the software was downloaded on several different machines (2 windows XP, 1 Windows 2004, 1 Mac running OS X). The installation is

very quick with the default destination of the install being c:\program files\openvpn. What makes one machine a client or server is dependent on how the configuration file is setup on each machine. OpenVPN can be run as a service on Windows XP so that it starts every time the OS boots up. It can also be run from the command line. The configuration file for OpenVPN residing in the ..\program files\config folder can also be used to start OpenVPN on a machine. Right-clicking on a configuration file which has the extension .ovpn for Windows brings up a context menu with the option to start a connection. By default starting OpenVPN opens a command window where one can see the messages being sent to the log file concerning the connection. This makes it very convenient to see what is going on and how the connection is progressing. A GUI for OpenVPN on Windows exists at http://openvpn.se/ which can be freely downloaded and used. This program, called OpenVPN GUI for windows, is also very easy to use.

In our laboratory set up (Figure 2), the IP address of the network is assigned dynamically. For a possible change in the networks IP address, a Dynamic DNS service found at https://www.dyndns.org/ can be used. This free service allowed picking a DNS name which is part of their domain. This DNS name is used in a "remote" statement on all client machines so that they are reachable even after the IP address change.

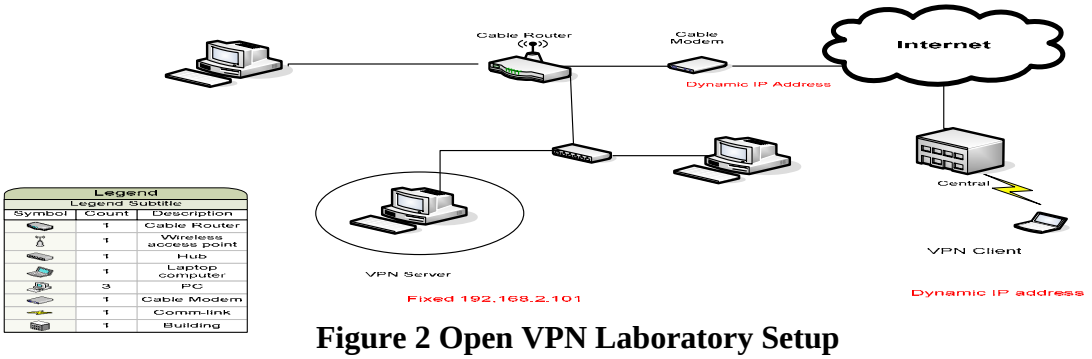

Figure 2 Open VPN Laboratory Setup

Laboratory Network Configuration

The laboratory network (Figure 1) consists of four machines (3 wired, 1 wireless). They are connected to the Internet and a Motorola SB5100 Cable modem. A cable router is connected to the cable modem which provides DHCP service to the machines on the network, a built in firewall and NAT to allow all machines on the Network to access the Internet. The cable router is a D-Link wireless router which provides both wired and wireless service to the machines on the network. Laptop will be used as the remote machine in the VPN and has 803.11g wireless. While the setup of the configuration file, part of the challenge is creating a firewall configuration that lets the VPN function and providing adequate firewall security. Firewall rules were added allowing the OpenVPN software to connect to the Internet. Other firewalls were not as accommodating such as TrendMicro's Internet Security and Zone Alarm 6 firewalls. Additionally a rule was needed to be set on the cable router's firewall to open up port 1149 and another to forward all inbound traffic to the server machine. For Routing configurations the VPN software requires a separate private network.

Static Key Configuration File

This static key configuration is very simple with only three lines [3]. The dev tun option, the ifconfig line, and the secret static key line are needed on the server machine. The remote machine needs an additional line which begins with "remote" as it has to make the initial connection to the server machine. Included on this line is the DNS name or the IP address of the remote subnet. Note also that the ifconfig line after the word "ifconfig" shows the VPN private address of the machine that the configuration file is on followed by the VPN private address of the machine to connect to. This type of configuration can only be used between two peers. This type of encryption is not as secure as the dynamic SSL/TLS system most often used with OpenVPN. See Figure 3 and 4 client and server configuration files for more details.

```
# Client configuration file     client.ovpn
dev tun
remote  vireya.dyndns.org
ifconfig 10.8.0.2 10.8.0.1
secret static.key
# enable LZO compression
comp-lzo
# moderate verbosity in log file
verb 4
mute 10
```

Figure 3. Simple Client Configuration File

```
# Server configuration file server.ovpn
dev tun
ifconfig 10.8.0.1 10.8.0.2
secret static.key
# enable LZO compression
comp-lzo
# moderate verbosity in log file
verb 4
mute 10
```

Figure 4. Simple Server Configuration File

Dynamic Key Configuration Files

An example of dynamic key routing configuration files [4]-[5] for a client and server can be seen in Figure 5 and 6. Besides demonstrating the use of dynamic keys, this setup allows multiple clients to connect to one server [4]. No private IP addresses are given in the configuration file, rather the IP addresses for the VPN subnet are assigned dynamically by the software to the clients. The "client" option designates a client machine while the "server" option designates the server machine.

```
# Sample Server mode ovpn file Demonstrated. TCP or UDP server?
;UDP is the default
proto udp
replay-window 512 30
dev tun
```

```
ca ca.crt
cert larry02.crt
key larry02.key
# Diffie hellman parameters.
dh dh1024.pem
tls-server
# Configure server mode and supply a VPN subnet. The server will take
# 10.8.0.1 the rest will be made available to clients. Each client will be
#able to reach the server on 10.8.0.1
server 10.8.0.0 255.255.255.0
# Maintain record of client <-> virtual IP address associations in this file.
ifconfig-pool-persist ipp.txt
# The keepalive directive causes ping-like messages to be sent back and
#forth over the link so that each side knows when the other side has gone
#down. Ping every 10 seconds, assume that remote peer is down if no
#ping received during a 120 second
keepalive 10 120
comp-lzo
# Persist options will try to avoid accessing certain resources on restart
persist-key
persist-tun
# Output a short status file showing current connections, truncated
status openvpn-status.log
verb 3
mute 10
```

Figure 5. Sample Dynamic Server Ovpn Configuration File

```
# Client ovpn file from OpenVPN2.05 Windows distribution
Client
dev tun
# Use udp
proto udp
replay-window 512 30
remote vireya.dyndns.org 1194
# Keep trying indefinitely to resolve thost name of the OpenVPN server.
resolv-retry infinite
# Most clients don't need to bind to a specific local port number.
nobind
# Try to preserve some state across restarts.
persist-key
persist-tun
# SSL/TLS parms.
ca ca.crt
cert larry_lptp.crt
key larry_lptp.key
```

```
tls-client
#Verify server certificate by checking certicate has nsCertType "server".
ns-cert-type server
# Enable compression on the VPN link.
comp-lzo
keepalive 10 120
ping-timer-rem
persist-tun
persist-key
# Set log file verbosity.
verb 5
# Silence repeating messages
mute 10
```

Figure 6. Sample Dynamic Client Ovpn Configuration File

Conclusion

This paper achieves both vendor based and open source tools by explaining the fundamentals of the VPN and laboratory exercises practiced in industry to the classroom. The laboratory setup mentioned in this paper fits our educational need for undergraduate students specializing in Computer Engineering Technology, Computer Science, and Networking Technology. In general students felt, OpenVPN is an excellent hands-on learning tool to understand the state-of-the-art network technology with minimal hardware and free software.

References

[1] Rajaravivarma, V, "Practical Studies of IP Security Virtual Private Network", ASEE 2005 Annual Conference, Engineering Technology Division, Portland, OR, June 2005.

[2] "OpenVPN Static Key Mini-How to", http://openvpn.net/static.html

[3] Rajaravivarma, V. "Open Source Virtual Private Network Experience in Classroom", 24th Annual Consortium for Computing Sciences in Colleges Eastern (CCSCE 2008) Conference, Frederick, Maryland, October 2008.

[4] "OpenVPN man pages – TLS", http://openvpn.net/man.html.

[5] "OpenVPN 2.0 How to – "Setting up your own Certificate authority (CA) and generating certificates and keys for an OpenVPN server and multiple clients", http://openvpn.net/howto.html#pki

Biography

Professor Veeramuthu Rajaravivarma is currently a faculty member in the Electrical and Computer Engineering Technology department at SUNY, Farmingdale State College, NY. Dr. Rajaravivarma has over 20 years of experience as an engineer, project director, program coordinator, and educator. His areas of research interest are Embedded Systems, Signal Processing, Computer Networks, and Security.