EasyOpenVPN

Kenny Hunter

Beesham Sarendranauth

Contents

Introduction

Platform Specifications

Source Code

Implementation

OSI Model Interaction

Demonstration

Conclusion

Appendix

Introduction

OpenVPN is an open-source software application the allows connection from point-to-point or site-to-site using a secure connection. It implements VPN techniques to create the secure connection needed. It uses a custom protocol that utilizes key exchange(SSL/TSL).

OpenVPN allows for peer authentication. That is, when connecting to another location, some sort of verification must happen. This is basically the pre-shared key that was generated when the OpenVPN server was setup.

A normal VPN client can be used to connect to the OpenVPN as long it has the client information needed by the server to authenticate.

Platform Specifications

OpenVPN can be installed on a Windows distribution or a Linux distribution but for the purpose of this study we will be using a Linux distribution(CentOS7/Debian).

Source Code

#!/bin/bash

clear

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "\* Welcome to EasyOpenVPN Installer \*"

echo "\* by: Beesham and Kenneth \*"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo " "

#checks if user is running file as root

if [ "$(id -u)" != "0" ]; then

echo "\*\*This script must be run as root\*\*" 1>&2

echo "\*\*Please log in as root then re-run script\*\*"

exit 1

fi

#Display Welcome Messages and Step by Step

echo "Hello, Welcome to EasyOpenVPN!, Sit back and relax during this setup!"

echo " "

echo "\*\*\*\*\*\*\*\*\*\*\*\* BASIC SETUP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "1) What is the IPv4 Address of the network interface you want to connect to?"

read -p "IPv4 Address: " IP

echo " "

echo "2) What port do you want OpenVPN on? (1194 is standard)"

read -p "Port: " PORT

echo " "

echo "3) Do you want to listen to port 53 as well?"

read -p "Port 53 (y/n): " OTHERPORT

echo ""

echo "4) Enable internal networking?"

read -p "Allow (y/n)? " INTERNAL

echo ""

echo "5) Enter Country Code (ex. CA, US)"

read -p "Code: " CONCODE

echo ""

echo "6) Enter Province/State (ex. Ontario)"

read -p "Prov/State: " PROVINCE

echo ""

echo "7) Enter City (ex. Toronto)"

read -p "City: " CITY

echo ""

echo "8) Enter Company Name (ex. Google)"

read -p "Name: " COMPANY

echo ""

echo "9) Enter EmailL"

read -p "Email: " EMAIL

echo ""

echo "10) What is your Client Name?"

read -p "Name: " NAME

echo ""

echo "GREAT!, we are done!, now we will install all the packages for you"

#Installing the Extra Packages for Enterprise Linux (EPEL) repository.

#This is because OpenVPN isn't available in the default CentOS repositories.

echo "Installing Extra Packages"

yum -y install epel-release

#Install OpenVPN.

#We'll also install Easy RSA for generating our SSL key pairs,

# which will secure our VPN connections.

echo "Installing OpenVPNServer and Easy RSA to generate SSL key pairs"

yum -y install openvpn easy-rsa -y

echo "Server Configuration Complete"

#Server is configured

#Generate keys and certificates.

#Easy RSA installs some scripts to generate these keys and certificates.

wget --no-check-certificate -O ~/easy-rsa.tar.gz https://github.com/OpenVPN/easy-rsa/archive/2.2.2.tar.gz

tar xzf ~/easy-rsa.tar.gz -C ~/

mkdir -p /etc/openvpn/easy-rsa/2.0/

cp ~/easy-rsa-2.2.2/easy-rsa/2.0/\* /etc/openvpn/easy-rsa/2.0/

rm -rf ~/easy-rsa-2.2.2

rm -rf ~/easy-rsa.tar.gz

echo "Creating directory for keys and certificates"

mkdir -p /etc/openvpn/easy-rsa/keys

#Change Directory into the easy-rsa directory

cd /etc/openvpn/easy-rsa/2.0

#Make it more easier

cp -u -p openssl-1.0.0.cnf openssl.cnf

#Set to 2048 bit encyption

sed -i 's|export KEY\_SIZE=1024|export KEY\_SIZE=2048|' /etc/openvpn/easy-rsa/2.0/vars

#Set Country Code

sed -i 's|export KEY\_COUNTRY="US"|export KEY\_COUNTRY="$CONCODE"|' /etc/openvpn/easy-rsa/2.0/vars

#Set Province / State

sed -i 's|export KEY\_PROVINCE="CA"|export KEY\_PROVINCE="$PROVINCE"|' /etc/openvpn/easy-rsa/2.0/vars

#Set City

sed -i 's|export KEY\_CITY="SanFrancisco"|export KEY\_CITY="$CITY"|' /etc/openvpn/easy-rsa/2.0/vars

#Set Org/Company

sed -i 's|export KEY\_ORG="Fort-Funston"|export KEY\_COUNTRY="$COMPANY"|' /etc/openvpn/easy-rsa/2.0/vars

#Set Admin Email

sed -i 's|export KEY\_EMAIL="me@myhost.mydomain"|export KEY\_COUNTRY="$EMAIL"|' /etc/openvpn/easy-rsa/2.0/vars

#Set Org Unit

sed -i 's|export KEY\_OU="MyOrganizationalUnit"|export KEY\_COUNTRY="$COMPANY"|' /etc/openvpn/easy-rsa/2.0/vars

#Create the PKI

. /etc/openvpn/easy-rsa/2.0/vars

. /etc/openvpn/easy-rsa/2.0/clean-all

export EASY\_RSA="${EASY\_RSA:-.}"

"$EASY\_RSA/pkitool" --initca $\*

export EASY\_RSA="${EASY\_RSA:-.}"

"$EASY\_RSA/pkitool" --server server

# Start creating the client keys

export KEY\_CN="$CLIENT"

export EASY\_RSA="${EASY\_RSA:-.}"

"$EASY\_RSA/pkitool" $NAME

# DH params

. /etc/openvpn/easy-rsa/2.0/build-dh

# Let's configure the server

cd /usr/share/doc/openvpn\*/\*ample\*/sample-config-files

cp server.conf /etc/openvpn/

cd /etc/openvpn/easy-rsa/2.0/keys

cp ca.crt ca.key dh2048.pem server.crt server.key /etc/openvpn

cd /etc/openvpn/

# Set the server configuration

sed -i 's|dh dh1024.pem|dh dh2048.pem|' server.conf

sed -i 's|;push "redirect-gateway def1 bypass-dhcp"|push "redirect-gateway def1 bypass-dhcp"|' server.conf

sed -i "s|port 1194|port $PORT|" server.conf

#Setupt the DNS

grep -v '#' /etc/resolv.conf | grep 'nameserver' | grep -E -o '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | while read line; do

sed -i "/;push \"dhcp-option DNS 208.67.220.220\"/a\push \"dhcp-option DNS $line\"" server.conf

done

# Listen at port 53 too if user wants that

if [[ "$OTHERPORT" = 'y' ]]; then

iptables -t nat -A PREROUTING -p udp -d $IP --dport 53 -j REDIRECT --to-port $PORT

sed -i "1 a\iptables -t nat -A PREROUTING -p udp -d $IP --dport 53 -j REDIRECT --to-port $PORT" $RCLOCAL

fi

# Enable net.ipv4.ip\_forward for the system

if ! grep -q "net.ipv4.ip\_forward=1" "/etc/sysctl.conf"; then

echo 'net.ipv4.ip\_forward=1' >> /etc/sysctl.conf

fi

# Avoid an unneeded reboot

echo 1 > /proc/sys/net/ipv4/ip\_forward

# Set iptables

if [[ "$INTERNALNETWORK" = 'y' ]]; then

iptables -t nat -A POSTROUTING -s 10.8.0.0/24 ! -d 10.8.0.0/24 -j SNAT --to $IP

sed -i "1 a\iptables -t nat -A POSTROUTING -s 10.8.0.0/24 ! -d 10.8.0.0/24 -j SNAT --to $IP" $RCLOCAL

else

iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -j SNAT --to $IP

sed -i "1 a\iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -j SNAT --to $IP" $RCLOCAL

fi

#Restart OpenVPN

if pidof systemd; then

systemctl restart openvpn@server.service

systemctl enable openvpn@server.service

else

service openvpn restart

chkconfig openvpn on

fi

#Test to see if you are in a NET Network

EXTERNALIP=$(wget -qO- ipv4.icanhazip.com)

#Test to see if the IP Matches, if not request it

if [[ "$IP" != "$EXTERNALIP" ]]; then

echo ""

echo "We have dected you're in a NAT, please type your external IP Address"

echo ""

read -p "External IP: " -e USEREXTERNALIP

if [[ "$USEREXTERNALIP" != "" ]]; then

IP=$USEREXTERNALIP

fi

fi

sed -i "s|remote my-server-1 1194|remote $IP $PORT|" /usr/share/doc/openvpn\*/\*ample\*/sample-config-files/client.conf

# Generate the client.ovpn

newclient "$NAME"

echo ""

echo "Finished!"

echo ""

echo "Your client config is available at ~/$NAME.ovpn"

echo "If you want to add more clients, you simply need to run this script another time!"

Implementation

OpenVPN can be installed to a linux distribution via a number of ways. One way was by following a specific number of commands that installs and setup the server. We took those commands and make a bash script so as to simplify the process. All that needs to be done is to run the script and enter the information needed/required/asked for.

The script will setup the server and generate the required files for the connection of a client.

The client will import the configuration file generated by the server in order to connect.

OSI Model Interaction

|  |  |
| --- | --- |
| Layer | Interaction |
| Layer 1 (Physical) |  |
| Layer 2 (Data) | SSL/TLS protocol |
| Layer 3 (Network) | SSL/TLS protocol |
| Layer 4 (Transport) | TLS protocol |
| Layer 5 (Session) |  |
| Layer 6 (Presentation) |  |
| Layer 7 (Application) |  |

Demonstration

Conclusion

Appendix

Project Proposal

Project Scope:

Project objectives: to create an easy, user friendly way to setup an OpenVPN server and/or client on a machine.

Goals: - create a bash script to aid in the setup of OpenVPN server

- create a bash script to aid int the setup of OpenVPN client

Tasks: - select specific OS to setup OpenVPN server on (client can be any OS)

- determine steps/commands to install and setup an OpenVPN server/client

- compile steps/commands into a server/client bash script

Resources: - CentOS7, Debian, OpenVPN software

Schedule: - Project time period: March 9 - April 26

- March 9: Project Overview

- March 16: Project Scope

- March 23: Project Research

- March 30: Project Development Begin

- April 06: Project Dev. Cont'd

- April 13: Project Dev. Debug

- April 20: Project Finalization

- April 26: Project Complete and Summission

Tools/Techniques/Technology used:

Technology used in the development of this project are:

* CentOS7
* Debian

Hardware/Software Specs:

* CentOS7 ver. 7.10
* Debian

Project Planning:

Schedule: - Project time period: March 9 - April 26

- March 9: Project Overview

This outline what the project is and should be about. To install/setup an OpenVPN server/client with a user friendly bash script is the main idea of the project.

- March 16: Project Scope

Determining the project scope is critical and helps to put the work entailed into perspective. This will help to distribute the work load per team member.

- March 23: Project Research

Project research is essential. Finding and determine the proper tools and technology to use to complete this project will aid in the efficiency of the end product.

- March 30: Project Development Begin

Project development begins.

- April 06: Project Dev. Cont'd

Continuation of project development.

- April 13: Project Dev. Debug

Finding any bugs in the code and creating patches to ensure smooth service delivery.

- April 20: Project Finalization

Finilization of the project. Proper documentation and user friendly introduction to how to use the service.

- April 26: Project Complete and Summission

There will be viva session for all students in first 90 minutes and demonstration session next 90 minutes.

* Project Proposal (Just attach to your report if you have already emailed it to me)
  + Tools/Techniques/Technology used:
  + Hardware/Software Specs:
  + Project Planning:
* Introduction of your project
* Tell us about platform used in your project (hardware/OS/Applications)
* Source code (scripts, instructions, methods, functions, references)
* Implementation steps / HowTo / Methodology
* Flow chart (display how your code or service is working)
* Create one table which has Layer1 to Layer 7 and add information relevant to your project
* Your report should be properly formatted and structured.
* I encourage creativity, feel free to add extra spice to your code.
* Put the snapshot of your steps in the report
* You should put every possible detail to prove that this work is original and done by you.
* If possible you can create a video demonstration and embed on your soft copy report (PDF/DOCX)
* Prepare and list down all the items you will demonstrate.