To: Steve VanDevender
From: Jerry Xie
Subject: Proposal for Approval for Final Project
Date: Aug 1, 2019

Hi Steve,

I have been working hard to envision an easy-to-maintain Virtual Personal Network, or VPN, Server for a long time, and I would like to request your approval for the project in order to continue innovating. For the past week, I have spent effort in investigating the best way to approach this project. Now I have come up with a really exciting project and hope you would share the same enthusiasm as me. The following is a brief proposal of the project including the scope, objective, and effect:

Summary and Motivation

This project is intended to deploy a self-maintained VPN server that can be connected from any modern personal computing devices such as laptops and phones. For security reasons, I switch my VPN server from one cloud provider to another monthly. It is tedious to repeat the VPN server installation and configuration each time. Though I have been using a dockerized OpenConnect Server container for a while, and it is smooth and robust, however, not every type of Virtual Private Server, or VPS, from cloud providers support Docker. Typically, one needs a Kernel-based Virtual Machine, or KVM, type of VPS to run Docker, and this kind of VPS is more expensive than others. For this project, I would like to try using Puppet to deploy a VPN server on Amazon Web Services, or AWS.

Tentative absolute Objectives

- Deploy a VPN server that is using any one of the following technology:
    - OpenConnect
    - Softether VPN
    - OpenVPN
- The VPN server is deployed on AWS.
- The VPN server can be connected from devices with iOS and macOS.
- Each VPN user has his own VPN user account credentials to connect.
- After the successful deployment, adding a new VPN user account should cost less than 5 minutes.

Tentative Optional Objectives

- Using Puppet to deploy and manage the VPN server.

Use Cases

A. Day-to-day usage on laptops/phones:
   Preconditions: I am connecting to a Wi-Fi network that is connected to the Internet, and I have the VPN server's address and my user account credentials.
    1. Ideally, I would be able to use the built-in network manager on the Operating System, or OS, to connect to the VPN server.

   PostConditions: After the successful connection, when I visit https://whoer.net/, I should be able to see my VPS's IP address instead of my actual IP address.

B. VPN user accounts updating usage:
   Preconditions: I want to perform an update, deletion, for example, a VPN account on the VPN server. I have the root access to the VPS hosting the VPN server, and I know the VPN account name I wanted to delete.
    1. I use SSH to connect to the VPS.

2. Ideally, there would be a text file stating all the VPN user account credentials; I delete the line defining the user I wanted to delete.

PostConditions: The account I deleted can no longer be valid to connect to the VPN server.

Deliverables

- A functional VPN server on AWS.
- A puppet module that deploys and manages the VPN server. (Optional)

Typical user population

For this project, the user population is mostly me, but it would benefit anyone with the need to use VPN services. If I succeed in using Puppet to automatically deploy and manage the VPN server, it would be flexible for consumers like me to switch VPN servers from one infrastructure provider to another in minutes without the pain of installing and configuring the software again.

Possible Security Issue

Since, in all of the three VPN solutions, the user account credentials are hashed in the config file, and the traffic between the client and server is encrypted, the only security breach is the administrator of the VPS. If we assume the administrator is a good citizen who does not take notes of VPN users' credentials, and keeps the software in the VPS up-to-date, there should be no catastrophic security issue for this project.

Life Cycle and Longevity

Ideally, if the optional objective is achieved, Puppet can ensure the whole system updated; otherwise, the administrator would have to do it themselves. I imagine there would be config files such as certificates and user-account-management-text-file to be maintained as time goes on and software updates. Other than these, the whole system for this project should be robust to use for years.

Documentation

An administrator's documentation stating how to install and run this project. If the optional objectives are achieved, then a programmer's documentation should be provided to explain the effect of code.

Project Plan and Timeline

- August 4 - August 5: Be familiar with building three VPN servers manually and pick one of them is the most feasible to be implemented for this project in one week.
- August 6 - August 11: Try integrating the deployment and management procedures with Puppet.