PERSONAL VPN SERVER USING RASPBERRY PI

A Project

Presented

to the Faculty of

California State University, Dominguez Hills

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Cyber Security

---

by

Brandon Ethan Mao

Fall 2021

PERSONAL VPN SERVER USING RASPBERRY PI

AUTHOR: BRANDON ETHAN MAO

APPROVED:

_____

Jane Doe, Ph.D.
Thesis [or Project] Committee Chair

_____

John Doe, Ph.D.
Committee Member

_____

Joe Doe, Ed.D.
Committee Member

This page is optional and should only be included if you intend to register your copyright with the US Copyright Office.

I want to dedicate not only this project, but my whole journey as a college student to everyone who has helped me learned and grow. May we all succeed with our lives and strive to be the best versions of ourselves. Now then, let’s get started shall we.

ACKNOWLEDGEMENTS

I would like to acknowledge all of the staff, faculty, and students that I have encountered, networked, and befriended throughout my time here at CSUDH. You all make spending a fortune here sound a little bit worth it, but not a lot though.

Also, shoutout to my fam and my two dogs Shadow and Coco. Apologies if the electric bill was high this semester. Look, OpenVPN is free and Raspberry Pis are cheap, but leaving a server on for 24/7 comes at a price.

PREFACE

I have no idea what to put here. But yeah, I talk about VPNs in this report. Go ahead and take a look now.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

VPNs allows end-users to sustain a private network connection while communicating over the Internet, adding extra layers of security over on the network. Many users may have heard of the benefits of using a VPN solely from advertisements of third-party companies. Companies would often utilize social engineering tactics to persuade the audience into believing their claims and purchasing the product. While some services can work efficiently, we are entrusting our purchases to third-party companies where the likelihood of data breaches, SPOFs, and EOSLs can happen at any moment. Throughout this project, I will primarily demonstrate how to create your own personal VPN server by utilizing inexpensive equipment and procedures including the Raspberry Pi 4 circuit board and open-source software. In addition, this report will also cover the history and commercial uses of VPNs, compare differences between services and personal servers, and go over any security and hardening methods.

CHAPTER 1

INTRODUCTION

Nowadays, it seems that many users are becoming wary about keeping their online activities safe. With the current state of the world still affecting millions of people today, many people are spending the majority of their time online. Whether we are surfing the web or conducting meetings virtually, safety among our online selves should also come to mind. One such feature that is relatively inundated among online users is through the use of a VPN.

To start off, let us ask ourselves this question: "What actually are VPNs and what are they used for?" A VPN is an abbreviation for a virtual private network. In essence, VPNs are one of the many features used as a layer of security for keeping data secured by encrypting any information being sent through any network. When a user enables a VPN tool, their current network will be rerouted to another server that is external. The VPN server can be located in another country due to the service's use of their geolocation feature. For instance, if I was using a public internet connection at a local coffee shop, I will be open to many risks thanks to the Wi-Fi's high availability. Risks can include chances of data being stolen or potential for a man-in-the-middle attack to take place. If I were to enable the VPN tool to a server set in another location (let us say a server connected in Spain), then my VPN will assist in acting as the middleman to ensure any data being routed in and out from my end is not out in the open.

VPNs sound like a simple tool to have for securing oneself. But this moves on to my concern: Which VPN service is the right choice, and is it better to have your own personal VPN server to yourself? Looking at it face-value, it seems that these VPN services can only be accessed by purchasing one of the many products that are offered among various third-party companies. It is also public notice that the only way to gain a VPN is by purchasing one yourself.

But it is never usually asked whether or not a user can create their own VPN themselves. And that is what I intend to do with my research.

The main points of the report will consist of the following key terms: research, create, analyze, and conclude. By researching, I will be looking over information relative to VPNs. This includes processes on how VPNs operate, guides on how to create your own VPN server, and VPNs being available to purchase from companies. By creating, I will focus my primary goal on creating my very own personal VPN server through the use of a Raspberry Pi microcomputer. To do this, I will be utilizing a free VPN software called OpenVPN and will be introducing more information about it on the upcoming chapters. By analyzing, this leads to my secondary goal where I would like to compare my results to those claimed by third-party VPN companies and showcase any differences or similarities that I was able to conduct. And by concluding the report, I will be recapping all of the events that played a role into the project development cycle and rundown several methods in which I can adjust for future work.

My expectations for this project would be that both a personal server and a third-party server would work fairly similar to each other, with a few caveats along the way. Given the current state of the world, I believe that operating something that is realistic to complete during this time should be my main focus. And because Raspberry Pis were available on my end, I believe working on this project will be my safest plan to showcase my skills in the field. Also, at the end of the project, I intend on operating the Raspberry Pi server for future purposes as a way to gain access to my home network from the public area.

In the next chapter, I will be going over the background of my topic and showcase several uses of virtual private networks. In addition, I will be describing my motivation into selecting this topic and addressing any problems or concerns that arise in this specific topic.

CHAPTER 2

BACKGROUND AND RELATED WORK

Virtual Private Networks were originally used for big businesses, larger organizations, and government facilities for keeping data secured. Specifically, this is catered towards employees working remotely online to avoid any potential data leaks, hijacks, or theft that can happen through wireless connectivity. A VPN tailored to a specific company will users to access internal data while still being on a safe network connection, even if they are accessing it externally.

There are many uses for commercialized VPNs. Some popular uses for a VPN is to access any region-locked content through the use of a geolocator feature and masking the IP address of a device. And of course, VPNs can act as an extra layer of security for your home network and setup. Popular VPN services currently available include NordVPN, Surfshark, and ExpressVPN – with each company having their own features and capabilities for their products. But what happens when the claims that are made end up becoming a false advertisement? For instance, do VPNs actual help keep your data private?[5] While VPNs are able to encrypt IPs and provide multifactor authentication, it does not protect users from attacks. In other words, users are still open to receiving potential cyber threats including the spread of viruses, malware, Trojans, bots, and spyware. At the end, user management with the VPN should also be a focal point.

Another example is that companies would like to boast about the performance speed of their products to compete with other businesses. Sometimes, these companies may be deceptive with their marketing and stamp a high number for their speed test in order to grab the user's attention for a faster connection. The problem with handling with performance and speed is that

every user will have a different outcome depending on many factors such as equipment setup and location – creating for different types of uploading and downloading speed.[6] Taken from a performance test made by ZDNet, the author, David Gewirtz, conducted his own comparison between the popular aforementioned VPN services where one of the sections was comparing the speeds.[3] A table of Gewirtz's results can be shown below.

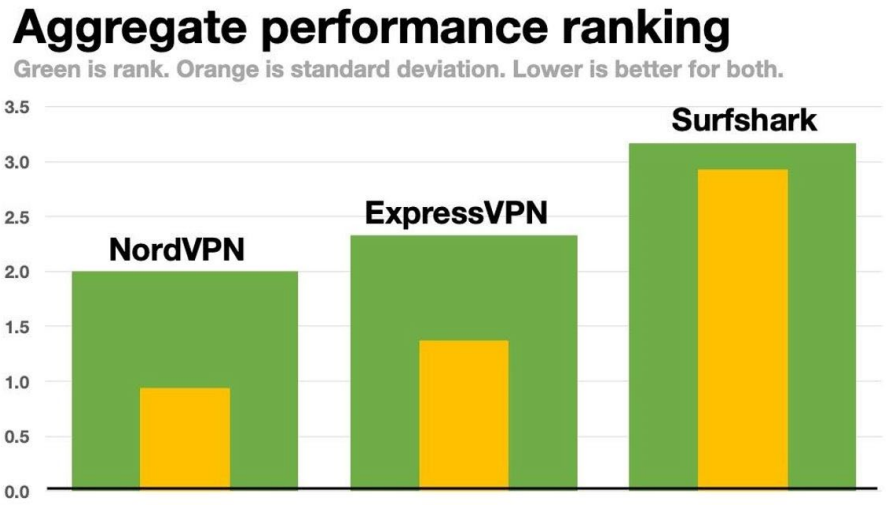


Table 1: VPN Speed Comparison by ZDNet

While having a VPN in general may boost the likelihood of secure connections, they are far from being the only security tool needed for every user. This is where misleading claims come in to play. For many companies, advertisements are created to spread awareness of their product to allow individuals who are interested to try out their service. This is the same deal for every company and is one of the ways on how they make a profit. The same goes for many of the third-party VPN services. Many commercials from popular VPN services can be seen on cable, magazines, or even the internet. Some of the claims that VPN companies advertises are often misleading. Claims such as how they utilize "military graded" encryption, keep activities hidden from ISPs, and being the only tool needed for security and privacy are very farfetched to be true.

You will see many of these claims being plastered over on the media. Many commercials are being aired displaying their products. Personally, I would notice that a lot of influencers or content creators over on sites such as YouTube or Instagram are being sponsored about these VPN services. What puzzles me about them are that these sponsors target channels who are most likely not in the realm of technology, so having these people tell me that I will need a VPN solely because it will keep me protected is far from the truth. For example, having a person who specializes in culinary all of a sudden just tells me to buy a subscription to ExpressVPN. It makes no sense; but from the viewers' perspective, the influencer will have high authority over their followers, making them listen to what the influencer will push out to the audience no matter what situation it is.

This leads me with the issues on how companies distribute their claims of the service through their social engineering tactics. Essentially, companies will utilize various strategies in order to deceive people to have them buy into their business. Just like the last paragraph mentioned, authority has the ability to enable a group of people into looking up to a public head or spokesperson relative to a certain topic and usually seems as they carry a huge understanding of what is asked or needed by the audience. Another trick that many companies, and not just VPN businesses, use is through the emphasis of urgency and scarcity – and sometimes both together. Urgency urges the users into believing their claims as they are a necessity to your livelihood. For this topic, advertisements for VPNs will always mention that you are unprotected from your computer and that using their product will allow you to hide your IP address. In a way, they are scaring you if you do not use their service. And as for scarcity, this usually happens on the company's website where a sale is up for their product and uses the "Get it before it is too late" motto that enables the user into rushing their payments.[11] In relation to this strategy, now

is a perfect time for companies to use scarcity as November rakes in the Black Friday deals for many products, and that include VPNs as well. Looking over VPN websites during the month of November, you will see huge phrases saying to purchase now all while having a countdown timer indicating that you have this much time before the sale ends.

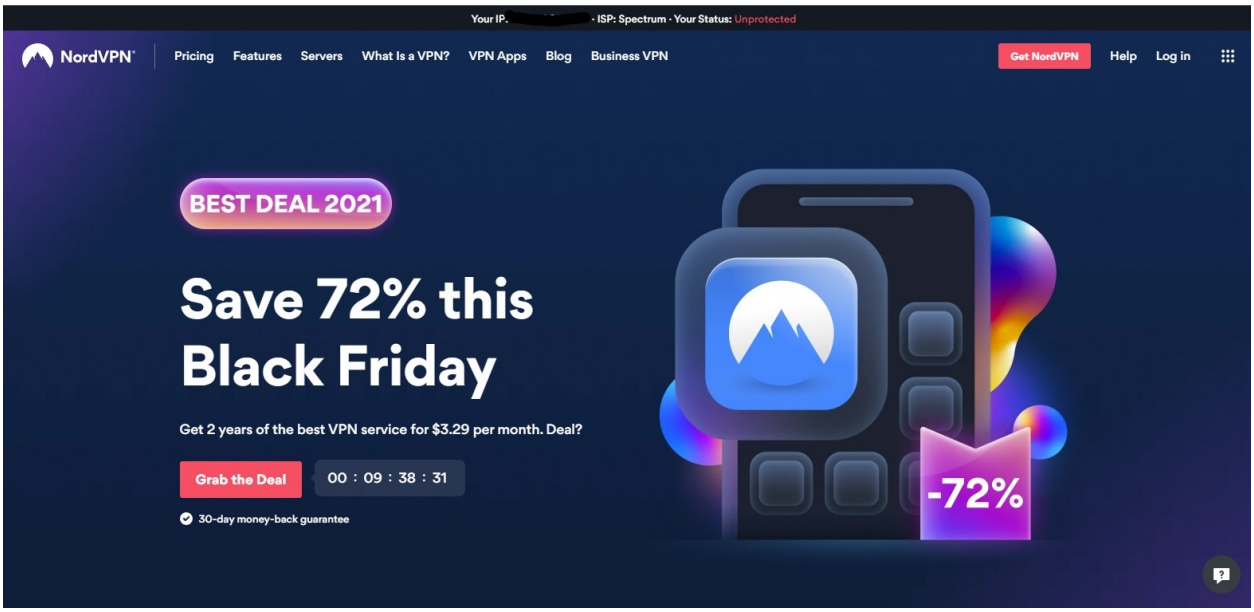


Figure 1: Example of Social Engineering from NordVPN

Aside from shady advertising practices, another issue for VPNs lies on the fact that not every VPN service is created equal. With the abundant need for cybersecurity, many startup companies and organizations are beginning to rise. This means that there are a swarm of VPN services that want a piece of the competition. But with numerous services to go through, what makes each one special compared to the other? Basically, why pay for service X when service Y offers more for the same price? Many comparisons between companies may overwhelm the user and will be bombarded with information. Obviously, at the end of the day, it is up to the individual's choice whether they decide to go with either product or not. And that is why for my

project, I hope that my reasonings and information will benefit those who wish to choose the personal server option and create a service for themselves.

The following chapter will go through the design options that I will be applying to my project. The methodologies will include two different types of design. The first design will rely on the physical aspect of the project where I will go over the setup of the Raspberry Pi microcomputer along with its respective components. The second design phase goes over the logistics and functionality of the software that is being used and how the Raspberry Pi server will be operated throughout this project. In addition, I will start to introduce the OpenVPN service beginning with how it looks from the surface. Through the aid of various guides[1][4][13], I will then demonstrate the installation process following the next chapter.

CHAPTER 3

PROJECT DESIGN

The project will consist of two aspects: the technical showcase where I intend to create the personal VPN server, and a comparison between the private server and a third-party service. Both factors will be distinguished further among the upcoming chapters.

The primary focus is on the VPN server that will take place on a Raspberry Pi 4 microcomputer being accessed with basic peripherals/KVM (keyboard, video, mouse). The default operating system installed is Linux, and will be configured by initiating Linux commands within the OS terminal. The Raspberry Pi computers have been used by many users as a safe platform to conduct various projects.

OpenVPN will be the main software used to assist with the server (https://openvpn.net/). The software is open-source and available readily for free, with the commercial Access Server version coming at a price. OpenVPN is a virtual private network system allowing for secure point-to-point connections among various devices. Upon further research, a version of OpenVPN is available to install on the Raspberry Pi OS through PiVPN (https://www.pivpn.io/). Essentially, it allows you to install OpenVPN, but adds in terminal commands and features to configure the VPN service more effectively and easily.

After successful configuration of the personal VPN server, documentation on its capabilities, tests with multiple devices and machines, and any known errors and downsides will be reported and compared with the other public services. My current devices and machines that will be tested includes a Windows 10 computer, a Macbook, my iPhone XR smartphone, and a VirtualBox virtual machine (VM) running an Android operating system.

The second half of the project will focus on the comparative analysis of several third-party VPN services and subscriptions. Comparison will include popular services such as Atlas VPN, ExpressVPN, NordVPN, and SurfShark VPN. Discussion will include social engineering/marketing strategies, subscription prices, and capabilities with other devices and machines.

Physical Design

The physical design of the project revolves around setting up the Raspberry Pi machine and powering it on. For this project, I am specifically working with the Raspberry Pi 4 Model B board with 8 gigabytes of RAM that I have purchased months ago prior to the graduate project course. I purchased the machine as part of a starter kit from CanaKit. The set comes with a 32-gigabyte Samsung EVO+ Micro SD card for storage that is already pre-loaded with NOOBS (New Out of Box Software) in which acts as a software that, when booted, will assist with the user to install the Raspberry Pi operating system and the rest of the features. It also comes with a system case, fan, heat sinks, Micro HDMI to HDMI cable, USB MicroSD card reader, and a power supply cord and USB-C PiSwitch (switch to turn the Raspberry Pi 4 on or off). The starter kit costs roughly $199.99 for the 8-gigabyte version; though if you wish to purchase just the Raspberry Pi board itself, prices can be as cheap as around $35. I have also purchased a few add-ons to customize my Raspberry Pi board a little further such as buying a new case, larger fan, and more heat sinks.

Physically, the Raspberry Pi will function as any computer or laptop would. This will including having to power on the system with their own power supply unit/cord, plugging in their display cable to a monitor, adding in any peripherals that are needed, and connecting it to a network for internet access. Below is how I have visually set up my physical methodology for

the project. To summarize what I have, the Raspberry Pi will lay flat on a table or flat surface as any machine would, and leave it plugged in with the power cable and switch connected. The power switch will allow me to easily turn the system on or off. Then I will need to ensure that full display is available. I used an HDMI cord connected to my smart television as a source for display. In the next section, I will go over another method to access the Pi system without using the Pi itself by utilizing SSH from another machine or device. Basic peripherals will also be used including a wired mouse and keyboard. Majority of the times, setting up configurations with the Raspberry Pi, and other computers, would just need a keyboard to navigate. To keep things simple, both peripherals have been used. And lastly, we will be connected the Raspberry Pi to my home network which is an important factor when implementing our VPN server. My current network service is from Spectrum, and I am using the SAC2V1K model for network connectivity. In addition, I am using an Ethernet cable to hardwire my Raspberry Pi to the home network, ensuring that my system is not using any wireless or Wi-Fi connection.

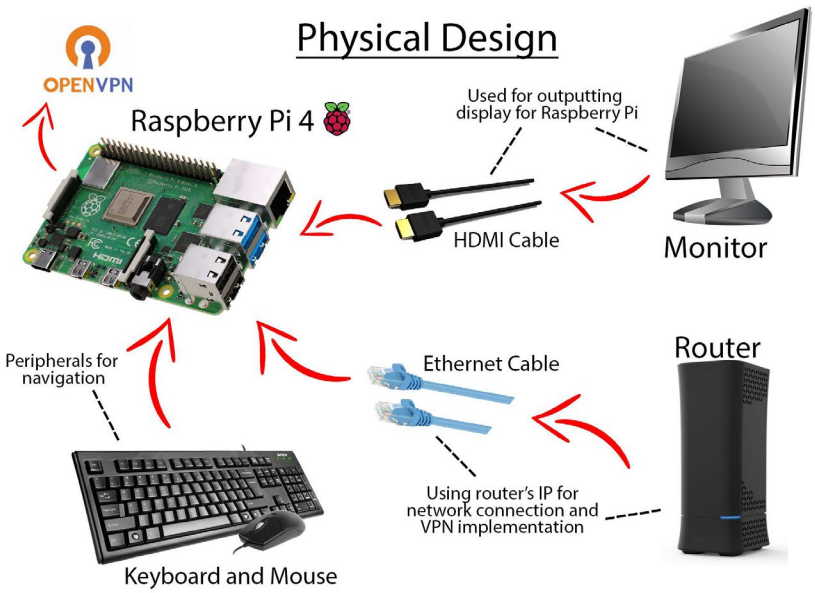


Figure 2: Physical Design

A quick note about Spectrum: To any users who currently use Spectrum as their Internet Service Provider (ISP), the company has decided to hard-lock the router settings page meaning that users are no longer able to access the router's configurations site on a browser of any machine or device. Instead, all configurations will be done through the Spectrum mobile application. This is to provide all services, including payments, television, internet, and phonelines, to all be under one domain, as well as making the router customizations to be accessible more easily as other routers would require you to enter the router IP address on the browser and enter a password to gain access. While I do think that having an application alternative is pretty beneficial, I would say that it created more problems for those who wanted to adjust through the computer system instead. Having a service to be forced through mobile devices only seems very counterintuitive, and removing the ability to access the router's settings on a computer is a bad call in my opinion. With that said, I am bringing up the Spectrum mobile application as it will be mentioned later on in the report as part of the VPN configuration process. Specifically, the application will be used for the port forwarding process of the implementation.

Function Design

Once the physical design is set, it is now time to distinguish the logically design of the project. In other words, we must think of how the server will operate and what must be done to create it, and this is where configuring the VPN service comes into play. While the Raspberry Pi itself is not free yet affordable, fortunately the software that I will be utilizing, OpenVPN, is open-source and is readily available for everyone. As already discussed, a version of OpenVPN called PiVPN will be installed exclusively for the Raspberry Pi. Implementation and installation of the PiVPN software will be heavily discussed in the next chapter. To summarize, a majority of

the installation process will be do through the Raspberry Pi's built-in terminal application. Entering a specific command will bring up the GUI for the PiVPN installation process. After successful installation of the software, the OpenVPN/PiVPN command will be available to use in the terminal. The command will function as the central hub in configuring and managing any user profiles, as well as conducting updates and backups to the service. Working with the VPN service requires you to transfer any created client profiles towards any of the machines or devices you wish to use for the VPN server. The client profiles will be created as their own files. So, for security purposes, it is best to transfer the client profile through a USB drive rather than sending the file through e-mail. This is to avoid sending out any sensitive data across the internet. When transferred, each machine will require the OpenVPN client software, which is available on most popular machines. When downloaded, users will be able to import the client profile in the client application, and connection to the VPN server should be running properly if done successfully. More information in configuring the files will be discussing throughout the testing and experiments chapter. A layout for the function design visual can be shown below.

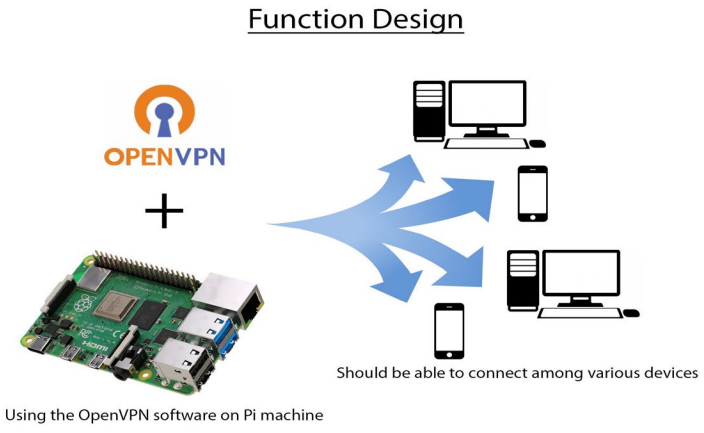


Figure 3: Function Design

Terminal Commands

Upon installing the PiVPN service, you will be given new commands to enter in the terminal. Entering the command *pivpn* in the terminal will list out various commands under the PiVPN tool. The list of commands include -a (add), -c (clients), -d (debug), -l (list), -r (revoke), -h (help), -u (uninstall), -up (update), and -bk (backup). These functions can be accessed by adding the flag after the *pivpn* command. For example, *pivpn -a* allows you to create a new client profile for the PiVPN service. A list of the PiVPN functions can be shown below.

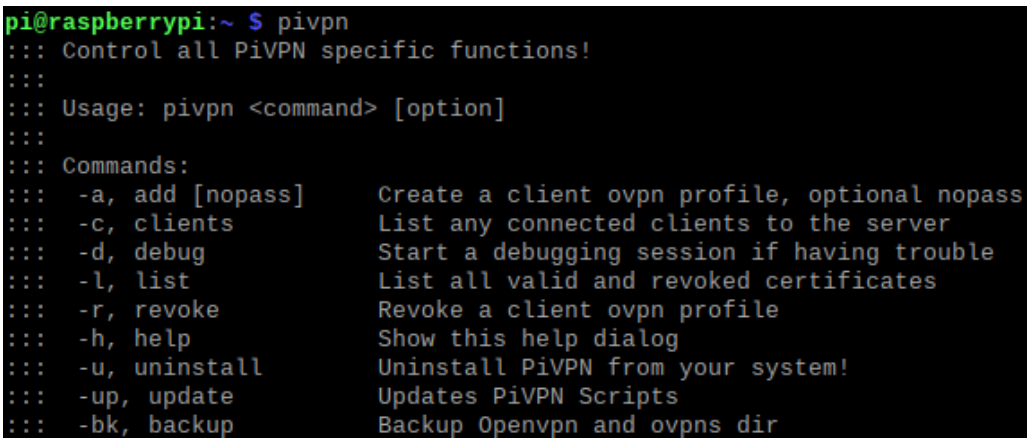


Figure 4: PiVPN Terminal Commands

To summarize each command, entering any of the flags after the *pivpn* commands would result in a specific function. Starting off, the -h command will act as your tutorial guide whenever you need any help. Sometimes OpenVPN may run into issues on your end. To troubleshoot it, you can try entering in the debugger mode with the -d command – essentially entering in a save mode to find any inconsistencies or errors. Updating the service can also help with the -up command, though automatic updates are usually on by default. If requiring any OpenVPN data to backup, you can enter the -bk command. And if you feel that you would like to

uninstall the OpenVPN service from the Raspberry Pi for any reason, you can do so with the -up command.

The *pivpn* command will also be your main source for adding in any new users or clients for your VPN server. For instance, the -a command will allow you to create a new OpenVPN client profile. Entering this command will list out some prompts for you to answer. Prompts includes entering the name of the client, how long will the certificate last, and creating and verifying the password for the client. The name of the file will be based on the name entered in the prompt. Once the prompts were successfully entered, the service will generate the client profile through using RSA encryption. Essentially, each profile will act as its separate key for accessing a specific client. Successful creation will save the client file from the following directory: home/pi/ovpns (directory may differ based on user settings). As mentioned earlier, it is best security practice to transfer the file through physical drive instead of electronic sending. Also make sure that each machine or device is giving their own client profile to avoid any potential conflicts.

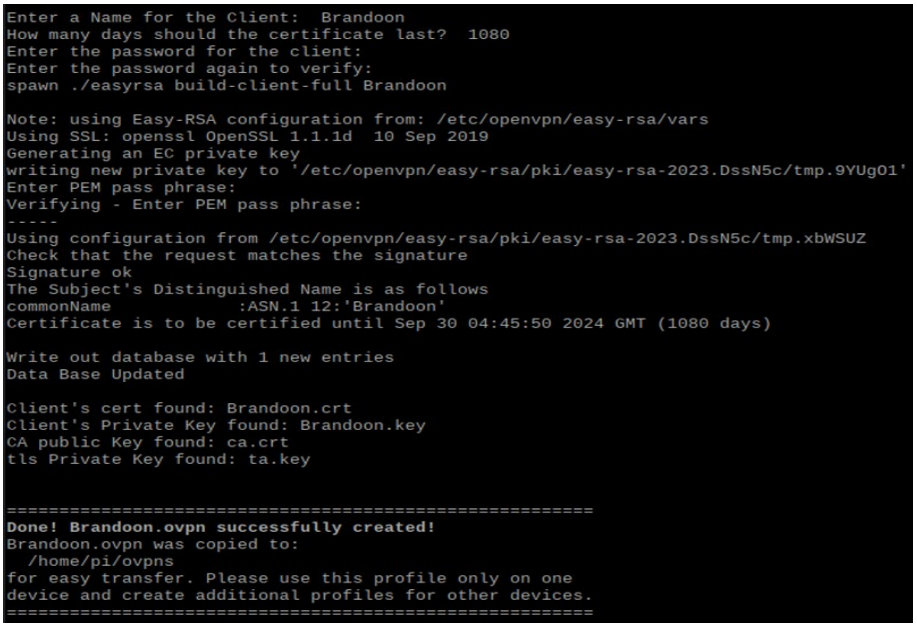


Figure 5: OpenVPN Client Profile Creation

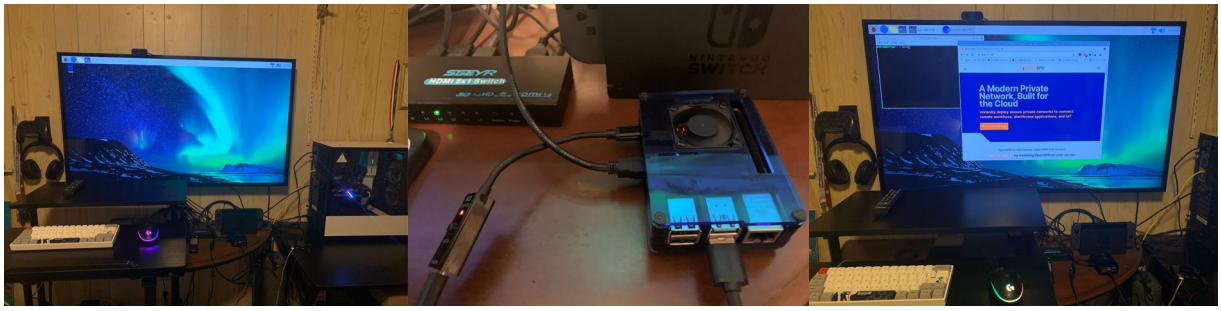
Other commands include viewing a list of clients. For instance, the -c command allows you to view a list of clients that are currently connected to the VPN server. This command will only display a list whenever any machine or device is connected directly from their respective client program, not if the machine or device is on. If there are no clients currently connected, the list will display nothing. If applicable, the list will display the names of clients currently connected, their remote IP address and virtual IP address, the number of bytes sent and received, and the date and time of when the client was connected. As opposed to the -l command which views a general list of valid clients that are created and those who have been revoked. The list will instead show the status of the client (either valid or revoked), the name of the client (as well as the name of the Raspberry Pi machine), and the expiration date of when the client's profile will expire. To revoke a client, you can enter the -r command and enter in the name and details of the client. Revoking a client will essentially make a client profile set to an expired status rather than a valid certificate. This means that the client file is no longer applicable for the machine or device to be connected to the VPN server. While I am using this server for myself, utilizing the revocation methods for larger scale organizations and companies will provide proper blacklisting of members to fit their specific access control policies, or in case any employees will need to be restricted of any access to a company's servers.

```
pi@raspberrypi:~ $ pivpn -l
: NOTE : The first entry is your server, which should always be valid!

::: Certificate Status List :::
Status      Name                                                Expiration
Valid       raspberrypi_175fd4f7-99aa-41d8-9758-e12f83b95dda    Oct 14 2031
Valid       Brandoon                                            Sep 30 2024
Valid       BrandonMac                                          Sep 30 2024
pi@raspberrypi:~ $ pivpn -c
: NOTE : The output below is NOT real-time!
:      : It may be off by a few minutes.

::: Client Status List :::
Name                Remote IP      Virtual IP      Bytes Received      Bytes Sent      Connected Since
No Clients Connected!
```

Figure 6: OpenVPN Listing Commands

Recap

Below are various screenshots on how I have physically arranged the Raspberry Pi. While not as organized or clean with the cable management, all required equipment were able to function as needed. As it shows, the Raspberry Pi 4 board is relatively smaller compared to your standard computer tower or laptop. The Raspberry Pi machine is about the same size as a credit card. Despite its smaller size, the Raspberry Pi works wonders when looking to create any project for newer computer or cyber security students. With its decent and cheap price, the Pi board be conducted equivalently to a sandbox machine to test out various settings or programs outside of your main computer. Prior to operating the Raspberry Pi to conduct my VPN project, I have previously used the microcomputer for other projects. For instance, I have borrowed a Pi board from the CSUDH IT department to development my own personal cloud storage server. Similarly, to my current project, the tasks requires scripts to be entered within the in-built terminal and an open-source software to assist with the service.



Figures 7: Hardware Setup

With that said, I believe that formulating the physical and function designs will ensure proper understandings of operating with the Raspberry Pi machine and OpenVPN/PiVPN software. Planning out my strategy for the project visually will help determine the methodology of setting up the server and knowing how to configure it. Setting up the design the way I did is

not only to assist myself, but to visually describe how each component operates for any new or upcoming user in the field. To recap my motivation for conducting this project, I wish to challenge these third-party VPN services - that utilizes social engineering and marketing tactics to persuade audiences into buying their products - into developing a VPN server of my own and access similar features from their own VPN products for free (or at an affordable price) and from the comfort of my own home. My expectations are that the OpenVPN server will be able to operate on every machine and device that I am testing with. Given that OpenVPN is open-source, I expect that the service will remain free to use and that my project will not cost anything, aside from the Raspberry Pi that I have purchased. And as I referenced Spectrum earlier, I was concerned whether or not I would have to access my router's settings page, which in a later part of the report will be the case. It is not an issue whether I know how to access it, but rather that the router page is strictly accessed on the mobile application and that features may have been removed.

For the next chapter, I will continue on with the project implementation and showcase the installation process for the OpenVPN/PiVPN service. While recent versions for the installation process have been given a GUI to enhance user experience, there are still aspects whe re an average user may still not understand what to select. To counter this scenario, the following chapter will go over each step in full detail and determine which option is best suited for the average user.

CHAPTER 4

PROJECT IMPLEMENTATION

The following section will go through the procedures on installing the OpenVPN software on the Raspberry Pi. Keep in mind that despite using a Raspberry Pi machine as the main system, any supported computer or cloud service are able to act as a main VPN server as well. The ability of having a service be made open source allows for many variations of a product to be made from many users. For instance, users who own an AWS cloud service may be able to conduct a similar project to mine without the need of any physical server. As mentioned previously, I am solely using a Raspberry Pi machine as it is a microcomputer that I have already purchased and owned prior to the project course, as well as demonstrating the use of an inexpensive device. As opposed to a cloud service, having a physical machine set as my VPN server would allow for full ownership and accountability to myself as relying on a cloud service would require you to have a sense of trust with a third-party.

Prerequisites and Setup

Before we can start with the implementation of the OpenVPN software, we need to ensure that our Raspberry Pi machine is set up properly. The prerequisites would be to setup the Raspberry Pi machine itself and plug in any of cables and peripherals including a display cable and monitor, mouse, and keyboard. For network connection, I am operating the system while being connected to the router with an Ethernet cable, making my system hardwired.

The Raspberry Pi that I have purchased is already pre-installed with the Raspberry Pi OS (also previously known as Raspbian). The operating system is Debian-based and provides the

primary foundation for operating the single-board computer. Basically, the operating system is equivalent to any Linux OS and most of the operations will be done through the terminal.

There are two main ways to access to the Raspberry Pi machine. The first method is to simply operate on the Pi board itself. The Raspberry Pi is essentially its own computer, allowing you to customize any configurations just like a normal computer would. If chosen this method, most of the implementation will be done straight from the built-in terminal application.

The second method is to SSH (secure shell) to the Raspberry Pi using an external device.[2] By default, SSH is disabled, so you will need to ensure that SSH is enabled on your Pi machine. Settings can be changed on the Raspberry Pi Configuration menu, or by entering the command *sudo raspi-config* then going to the Advanced Options menu. While the Pi machine is on, users are able to remotely access the Pi on any device or machine that supports remote/SSH connection. For Mac and Linux users, using the built-in terminal application is all you will need to connect. For Windows users, multiple third-party services offer SSH capabilities. An application I recommend is PuTTY, a free software that is used for simple SSH connections. There are various SSH applications available on mobile devices, though it is best practice to operate on machines connected to networks through wired means and avoid Wi-Fi connections to limit the chances of man-in-the-middle appearances. Regardless of the device or machine, the command to access the Pi server is usually written as:
ssh pi_username@local_address, with *pi_username* being replaced with your Raspberry Pi machine's name, and *local_address* being replaced with the IP address of the Pi (current IP address can be found with the *ifconfig* command). The default username for the Raspberry Pi is named as "pi", with the default password being "password". For best security practice, ensure that the default username and password is changed as the default usernames and passwords are

the same for all manufactured Pi machines. Changing them will likely decrease the chances of any brute forcing attempts. Once the SSH command is entered, enter the Pi's password and connection should now be access. Successful SSH connection will allow you to conduct any terminal commands straight to the Raspberry Pi machine.

Method two may be difficult to some users new to the field. To keep this project more digestible, I have decided to go with option one and operate on the Raspberry Pi machine instead. This will allow me to fully go in-depth with the implementation as I am operating on the Pi itself. Regardless of which method you decide for accessing the Raspberry Pi server, functions and commands will remain the same.

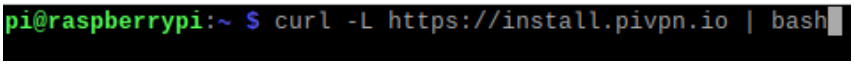
With the Raspberry Pi machine all set up, it is good etiquette to first perform updates to the system. To start, you will need to open up the terminal application and enter the command: *sudo apt update* and then *sudo apt full-upgrade*. These two commands together will seek out any packages that is available to update and downloads them from their respective sources.

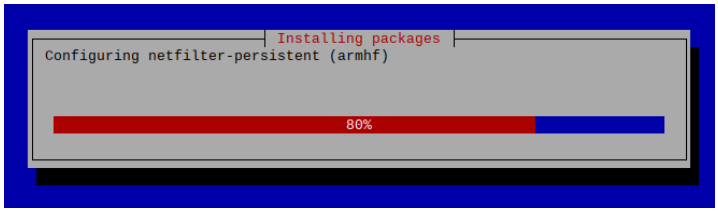
PiVPN Installation

Once everything is set up, it is now time to install OpenVPN. On the Raspberry Pi, the OpenVPN is installed through a GUI windows called PiVPN. Successful installation will grant users to utilize various settings for configuring OpenVPN clients. Back then, installing PiVPN would require many commands to be entered through the terminal. Nowadays, installation for the software is now simple and can be started with one command: *curl -L https://install.pivpn.io | bash*. This command will essentially grab the installation data from the PiVPN server and automatically start the installation process straight on the terminal.

Figure 8: PiVPN Webpage

To be more user friendly, the installation now aids the user with a GUI to guide them with the process. Once entered, the PiVPN installer GUI will pop up and will greet you with the message “This installer will transform your Raspberry Pi into an OpenVPN or WireGuard server!”, stating that installation of either service is about to begin. As a reminder, we will be installing OpenVPN instead of WireGuard for this project. When you are ready or satisfied with each step, select Ok by hitting the Enter key to continue. Keep in mind that installations may differ for other users. The steps here will solely be based on my configurations, but I will address when settings may be different.





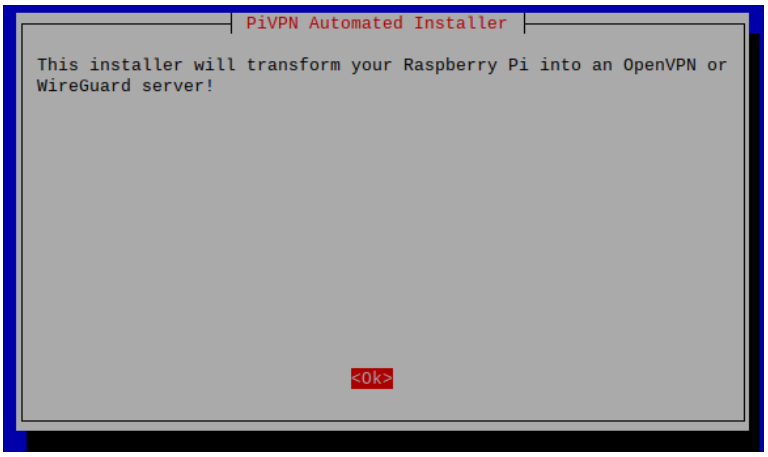
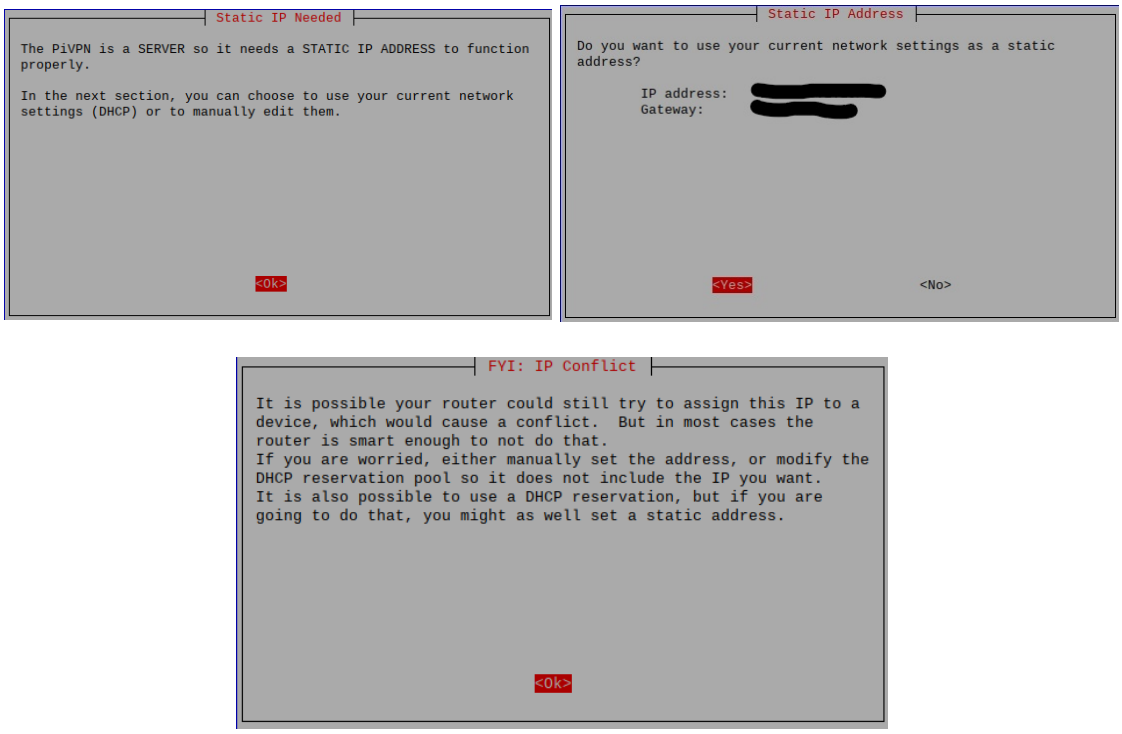


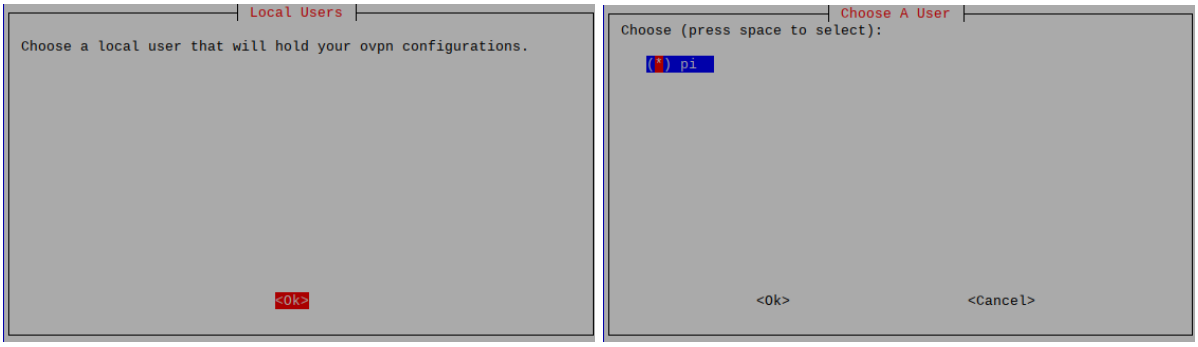
Figure 9: PiVPN Installer Startup

The next step will assure you that the server will need to be done with a static IP address in order to function properly. Static IPs mean that IPs for a device are assigned and are no longer changed, whereas dynamic IP addresses are frequently changed, resulting in different IPs to be assigned for the device. It is important that the server is set with a static IP address, so devices are able to connect to the Pi and avoid any conflicts if the server side has been changed. On my end, my IP addresses are static and will not change. The following step will confirm my current network settings will be used as the static address. The next step addresses the IP conflict where the router may try to assign an IP to the device; but for most average users, it should not be concerning as the router is smart enough to not do that. For users operating with dynamic IP addresses, they will either need to purchase a router that enables static IP addressing, or purchase a custom DNS from a dynamic DNS service that will be used aside from your current IP address. You can purchase a custom domain from various DNS hosting services. Because I am limiting myself for this project in terms of cost, I will go on with using my static IP address instead.



Figures 10: Static IP

The following step will ask the user to select the local user that will carry all of the OpenVPN configurations (client profiles). For best practice, it is good to have another user that holds all of the config files rather than relying on one administrator account. For this project however, I have left it with the default user “Pi”.



Figures 11: User Selection

Afterwards, I have been asked to select the VPN I wish to install. The choices are either OpenVPN or WireGuard. As described from the window, WireGuard is a fairly new VPN service that features faster connection speed, high performance, and modern cryptography. It is also mentioned that WireGuard is a preferred choice if you want to utilize a VPN on mobile devices as it provides efficient battery life compared to using OpenVPN. For this project, I will be installing OpenVPN instead as it keeps the more simple and traditional VPN methods for average users.

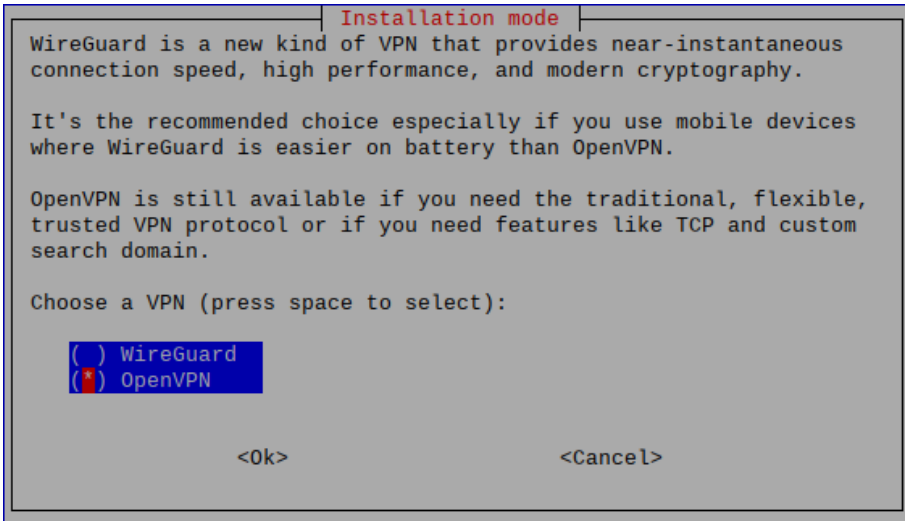


Figure 12: VPN Selection

Moving on with the installer, the next window will tell us a view settings that OpenVPN would keep as default for most users, as they are frequently selected for the average userbase. Such features include protocol choice, custom domain, and encryption. To keep the defaults, the user can select Yes with the Enter key. However, I will select No to showcase the features fully by going through each setting manually.

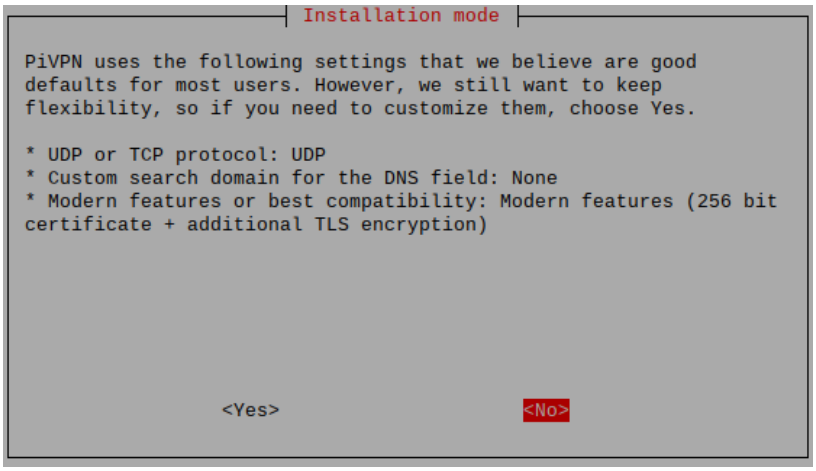


Figure 13: Default Settings

Starting off, we will need to select either UDP or TCP as the protocol. As default, OpenVPN recommends using UDP as it helps connect with the VPN server much quicker than TCP. Though you are still able to operate under TCP if you choose to do so.

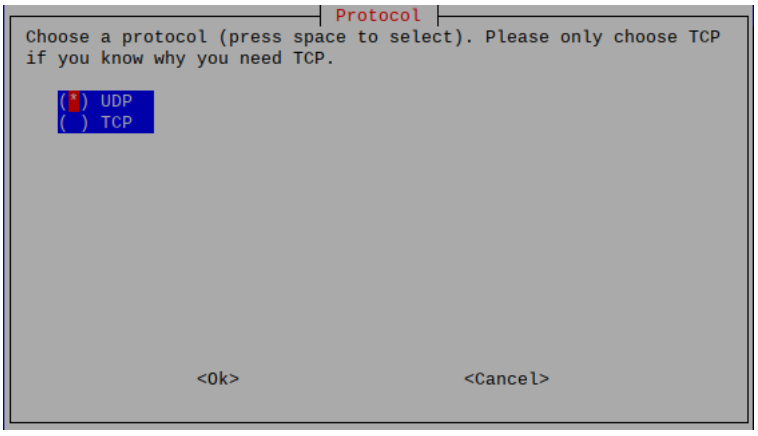
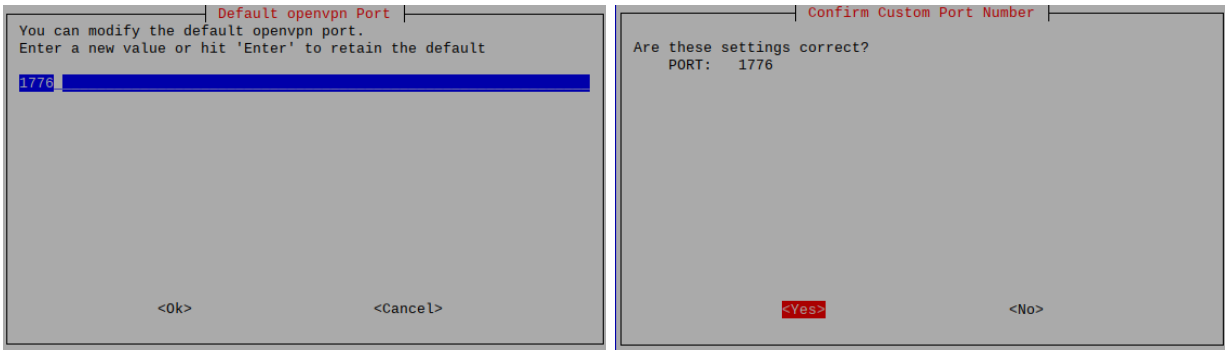


Figure 14: Protocol Selection

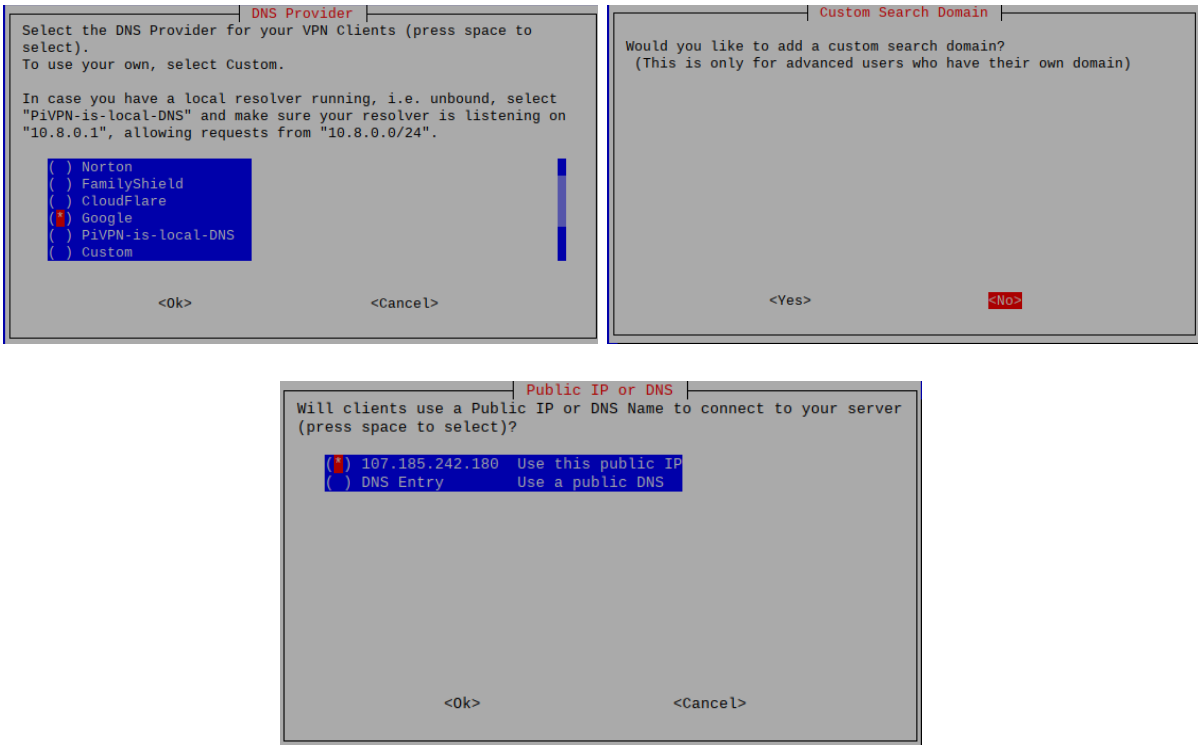
Next, we are asked to select the port number we would like to assign the VPN server for. By default, the port number is set to 1194. The choice of the port value varies on your own decision as long as the variable is higher than 1020 and is currently not being used. For my

project, I have set the port number to 1776. Keep in mind that after installing the OpenVPN software, we will need to go over to our router's settings page and port forward the number that we have assigned the server to. Confirm that the number is the correct value you want for the server and move on.



Figures 15: Port Number Selection

Following the port number creation comes selecting the DNS provider for your custom domain. This usually applies to users who have their own domain and would like to use it for the VPN server, or those who are operating with a dynamic IP address. Since I am not going to use a custom domain, select any option listed (I have selected Google for now) and select Ok to go to the next option that asks if you would like to add a custom search domain. If you would like to use a domain, select Yes. Since I am not interested in purchasing a domain for myself, I have selected No and continued on. Here, we will be asked if we will be allowing clients to use a public IP or a DNS name when connecting to the server. For a larger scale company, it is probably beneficial to utilize their own domain. As for myself, I went along with using my own public IP address.

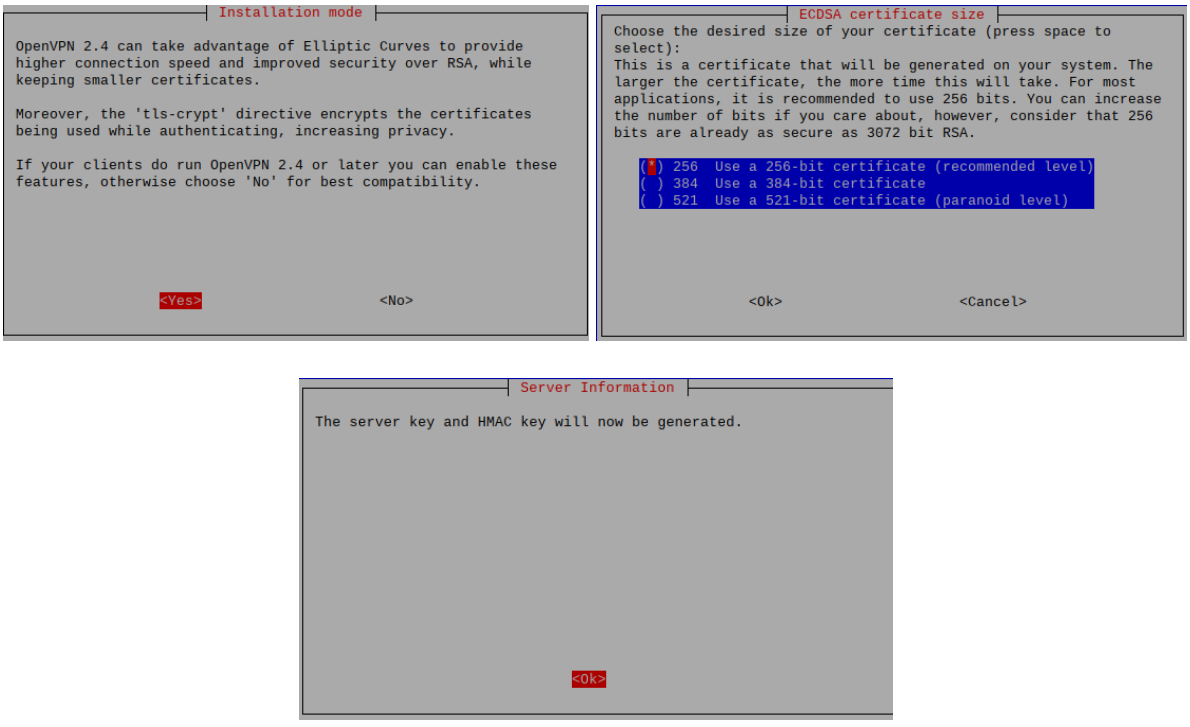


Figures 16: DNS Selection

Since we are in the cybersecurity field, we cannot talk about a topic without mentioning their security posturing techniques. For instance, PiVPN goes through the various security tools that the user has the ability to enable, starting with the option to operate with Elliptic Curves to provide higher connection speeds and increased levels of security over RSA. Turning it on will provide encryption for certificates while the authentication process, but will also limit the range compatibility. Afterwards, you will then be asked to select the size of your certificate. The higher and larger the certificate is, the more time it will take for connecting. The available choices are 256-bit, 384-bit, and 521-bit, with the option to increase the number of bits if you choose to do so. Other than that, OpenVPN recommends going with the 256 option as it is usually enough for the server to work. As they mentioned, 256 bits are just as secure as operating with a 3072-bit RSA encryption. However, any option will do. Once selected, the server key and HMAC key

will then be generated. Keys will be saved under the OpenVPN directory of the Raspberry Pi. Aside from certification and encryption, most of the security practice relies on efficient management and user controls.

There are more ways to implement security after installation has been completed.[8][9] In hindsight, majority of the time spent with the server is simply managing them with the utmost policy depending on the business. As mentioned, managers are able to customize the RSA key size if a higher encryption rate has a big priority for a company. Additionally, ensure that your root and administrator accounts are kept secured, and that other users are given their own user account separate from those that are set with high priority rules.



Figures 17: Encryption Selection and Key Generation

Because the mindset of operating a VPN server is to keep it on 24/7, we would want the server to operate on its own with the least interaction. Thus, we have a choice to enable

unattended upgrades which will allow OpenVPN to automatically install security upgrades and patches to the Raspberry Pi server. By default, it is usually set to on. After that is done, the PiVPN installer comes to an end. As mentioned from the last chapter, successful installation will provide the user with the *pivpn* command which allows users to operate the OpenVPN server functions through the Pi's terminal program. The command comes equipped with their own abilities such as the command to add new client profiles.

Figures 18: Final Installation Settings

For the next chapter, I will go over how to transfer the client profiles to the different types of machines and devices that will be tested on for this project. In addition, I will also go through the configurations and errors that I have encountered while experimenting this semester. I will then end the section by finalizing the report with a comparison between my personal VPN server and third-party VPN services.

CHAPTER 5

TESTS AND EXPERIMENTS

Once installation has been completed, experimenting the server commences. The devices that I have tested the OpenVPN service includes a Windows 10 machine, Macbook laptop, iPhone mobile device, and an Android operating system conducted from a virtual machine (VM).

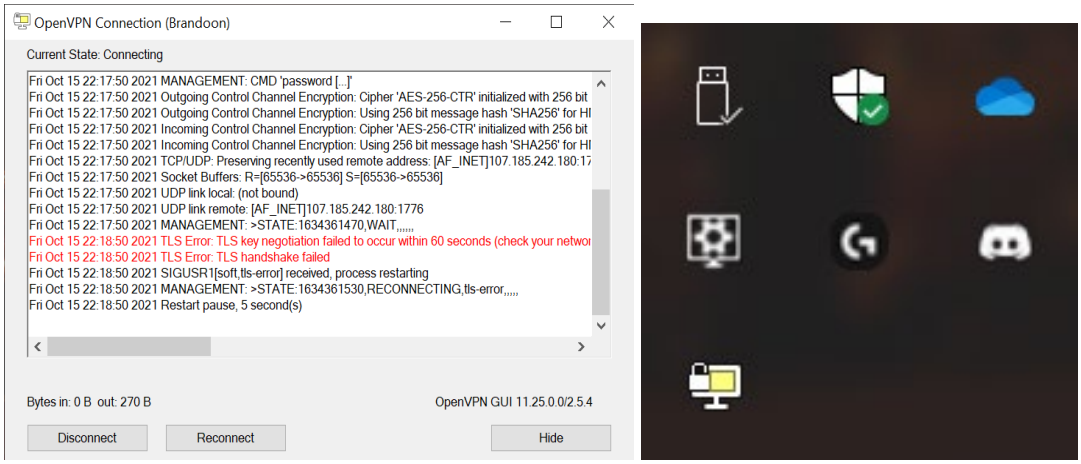
Configurations and Errors

Before going through the results of each domain, here are some possible configurations that you may need to be wary of, as well as some errors that I have encountered along the way.

First off, you will need to ensure that you have port forwarded the port number that you have assigned to use for the VPN server. In this case, the port number that I have decided to use is external and internal port 1776. As a reminder, the default port number for the OpenVPN settings is 1194, with the recommended number to be any available port above 1020. Port forwarding can be done within your router's settings page. In my case, configurations will be done through my Spectrum mobile application under its respective port forwarding feature. Also, ensure that the protocol for the port is assigned to UDP (or TCP if assigned differently) to match the configurations made from the installation.

This may not affect all machines, but sometimes the firewall rules can also blacklist the OpenVPN client connection. For some, this will not allow the VPN to not connect properly. To counter this, it is a safe bet to set up firewall inbound and outbound rules to allow our VPN server to connect with our machine. Configurations I have made revolves around creating new rules to allow any traffic from port 1776 to be whitelisted. Keep in mind that not all machines or devices are required to do this. But it is a safe keeping ensuring full connection with the server.

A final error that I have encountered is trying to access the VPN server all while being connected to the same network as the server itself. For example, my first test connecting my Windows desktop to the VPN server was having connection issues as the logs reads that a TLS handshake error was conflicting.[9] This was due to the machine being wired as the same network as the Raspberry Pi, which is my main home network. Having a machine connect with the same network was creating a collision between the client and the server. To counteract this situation, I had to use a USB wireless/Wi-Fi adapter, which I have owned prior to the course, to gain access to external wireless connections (does not apply if machine has built-in Wi-Fi). For testing my Windows computer, I connect it to my smartphone's hotpot to act as an external network. With that, the errors are no longer present when connecting to the server.



Figures 19: Error Signs

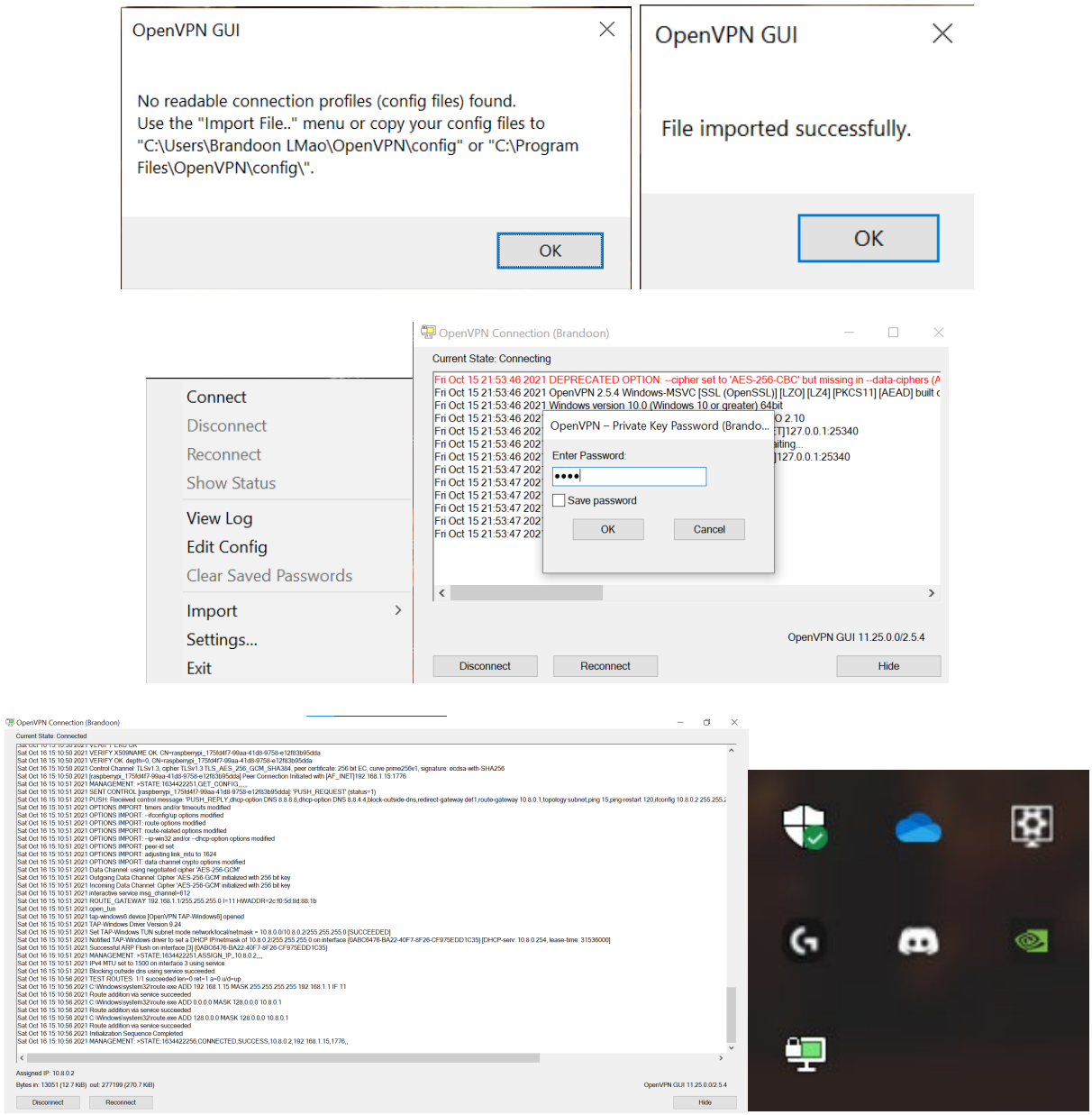
Windows 10

For the Windows 10 system, I am testing it on my personal computer that I have built months prior to the project course. Since this is my main desktop computer and is physically connected to the same network as the VPN server, I would need to connect to a different network connection to avoid any collisions and conflicts. To fix this, I utilized a USB Wi-Fi adapter to

pick up any wireless signals and connected to my smartphone's hotspot connection instead. Because of this, performance speed was relatively slow but was still able to operate properly.

OpenVPN offers a client connection software for users to connect to their VPN server. The application is called OpenVPN Connect (https://openvpn.net/client-connect-vpn-for-windows/) and is available in most supported devices. Successful installation grants you the OpenVPN client GUI and tools to connect to your VPN server. Clicking on the application will bring up an OpenVPN icon onto your taskbar (the icon is a computer monitor with a lock on the left side). Right-clicking the icon allows you to perform various functions such as connecting or disconnecting to the server, displaying the status or logs, or editing the configuration file of the client profile. It also allows you to import your client profile either through a URL, OpenVPN's premium Access Server feature, or through your own files. I went with manually importing my file that I have transferred over from my Pi board. When importing and connecting to the server, the password that I have created for the client is required. After entering the correct password, connection to the server should take around 5-10 seconds. Successful connection will result in the OpenVPN taskbar icon to be green. When trying to connect or having issues, it will be yellow instead. Regardless, connection to the VPN was a success. The name given ot the Windows client is "Brandoon".

As mentioned previously, I was having several errors when trying to connect to the servers. For starters, make sure that your router is port forwarded to the correct port number that you have enabled for the OpenVPN server. Then make sure that firewall rules do not conflict with any of the VPN operations. Finally ensure that the network you are currently on is not the same as the one used for the VPN server.



Figures 20: Windows 10 Test

MacOS

For testing on a Macintosh operating system, I will be using my personal Macbook that I have purchased years prior to the project course. The OpenVPN client application is also

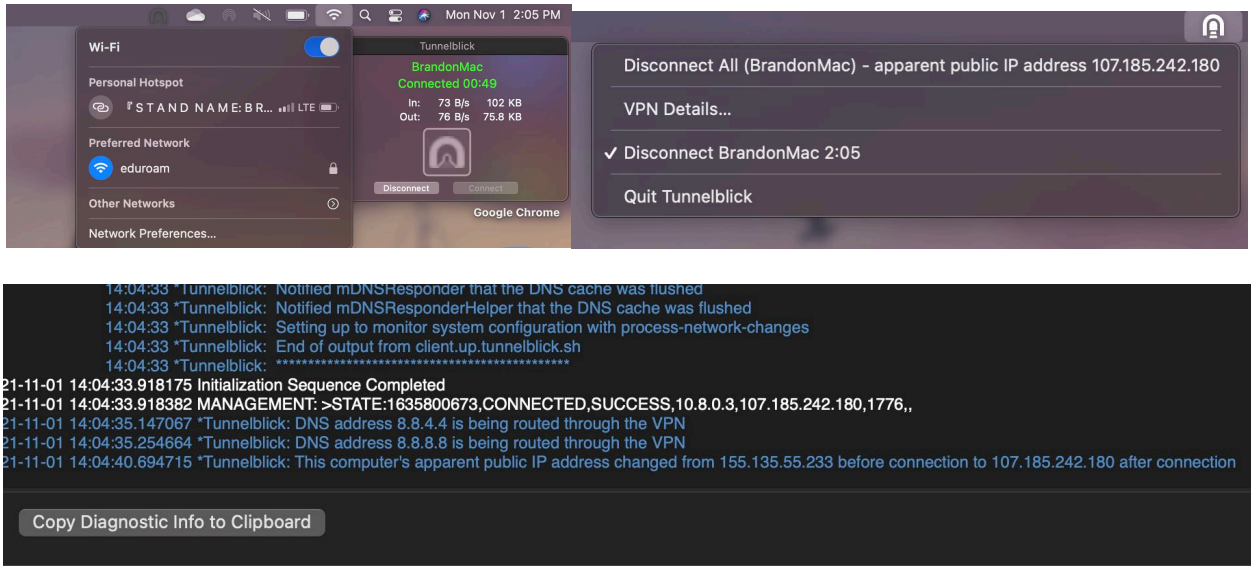
available for Mac computers. But instead of using the client, I have decided to utilize another third-party VPN client program just to test how they differ from the original program.

The application I have used for the Macbook is called Tunnelblick, another free and open-source alternative for connecting to VPN clients. Tunnelblick supports connecting to OpenVPN clients exclusively on MacOS stations. Downloading the application is simple as visiting the Tunnelblick website (https://tunnelblick.net/) and selecting the current download. Run through the installation process, and much like Windows, the Tunnelblick program will be downloaded and can be viewed on the toolbar/taskbar of your computer. Though unlike OpenVPN GUI, opening up Tunnelblick will display a more detailed window, coincidentally enough. To import a client profile, ensure that the file has been transferred securely on the Mac and simply drag and drop the file over on the Tunnelblick icon on the top right of the screen. This will automatically add the client profile to the system. When trying to connect to the VPN server, it will ask you to enter the password you have created for the client profile. Enter it and connection to the server should be attempted. The name given to the MacOS client is "BrandonMac".

The networks that I have tested my Macbook on includes my smartphone's hotspot service and the campus Eduroam connection. Between the two connections, Eduroam was able to perform well with the VPN enabled as I am utilizing an already established network domain rather than cellular data. Connection away from my home network works surprisingly fast and connecting to the OpenVPN client usually takes around 5 seconds.

Much like the Windows testing, I was having issues connecting my Macbook to the VPN server. This is due to the network dilemma again where I am not able to be on the same network

as the VPN server to connect to it, as I have made the server be made on my home network. As stated, testing with the Macbook has been done externally away from my home network.



Figures 21: MacOS Test

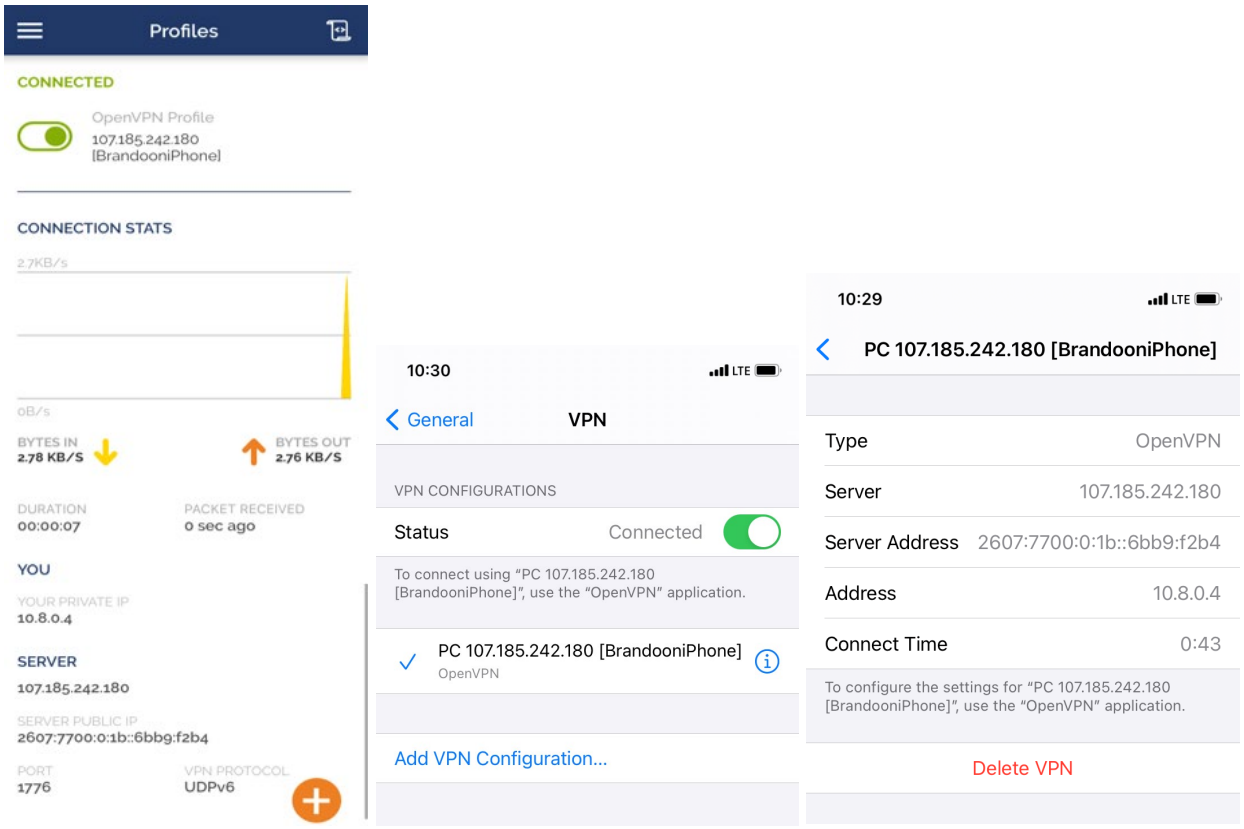
iPhone (iOS)

For testing on mobile devices, I will be conducting on my iPhone XR smartphone purchased years prior to the project course. To cater towards mobile users, the developers of OpenVPN has created a client connector application to work with mobile devices (smartphones and tablets) called OpenVPN Connect. For iPhone users, OpenVPN Connect can downloaded for free over at the Apple App Store. While for Android users, the client application can be found in the Google Play Store.

Once downloaded, the application will be available on your dashboard. Selecting the program will direct you to choose how you would like to import your client file. The choices are by URL domain, which I am not using, and file. Because iPhones do not have a USB port, our options are to send the file through email, which is deemed as unsecured, or through iTunes

syncing. I went with the iTunes method though the process was pretty lengthy. To start off, I will need to drop a copy of the client file from the Raspberry Pi onto a USB storage drive and then deliver it to any machine containing iTunes (I went with my Macbook). After the file is on the machine, I then connected my iPhone to the Macbook and opened up the iTunes program to start syncing the files. While tedious, I was still able to get the client profile up on the iPhone application. When the file is imported, it will ask you to enter the password that you have created for the client. Afterwards, the profile should be able to be added to your device. Connecting to the VPN server usually took around 1-2 seconds. The name given to the iPhone client is "BrandooniPhone".

Networks that I have connected to includes my cellular data/LTE service, public Wi-Fi, and the campus Eduroam domain. With no surprise, the Eduroam network offers better performance speed as it is an established domain that also requires DH authentication for verification and security. Working with my cellular data or public Wi-Fi connections were not too slow but took up some speed compared to when I am not connected to the VPN. Despite that, having a connection that routes back to my home network is extremely beneficial whenever I am connected to an open network where many users are connected to it.



Figures 22: iOS Test

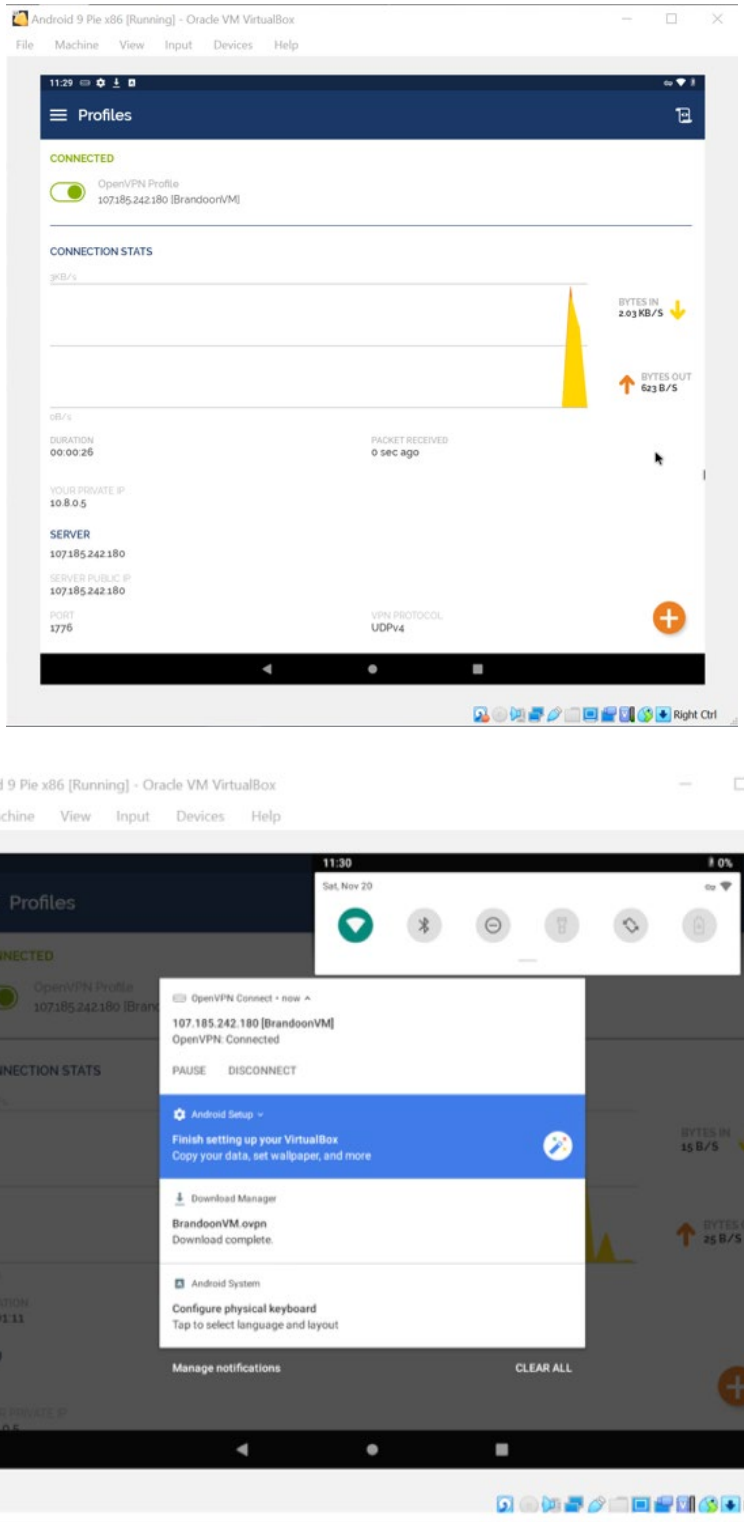
Android OS (VM)

In addition to testing on machines and devices, I also wanted to test the client profile over on a virtual machine. With Windows, MacOS, and iOS already settled, I have decided to go with an Android operating system to test the VM on. The Android operating system I am working on the virtual machine is the Android 9 Pie x86 version. The virtual machine will be conducted through VirtualBox and is being hosted over on my personal Windows computer. Installing the Android OS on a virtual machine was pretty simple and mostly consisted in configuring the account settings.

Much like the iPhone, the Android VM will require to download the OpenVPN connect application over on the Google Play app store. Once downloaded, the OpenVPN application will

be available over on your dashboard. The GUI of the program is relatively similar to the iOS version; though since the virtual machine is running through my computer, I had the ability to easily transfer the OpenVPN client profile through a USB drive rather than having to connect to iTunes to transfer. With that, importing the file is the same as the iPhone version. So once the client file is selected, the client profile has been added and is able to connect to the VPN server. The name given to the VM client is “BrandoonVM”.

Since I am operating the virtual machine at home, I am only able to connect to my iPhone’s hotspot to be connected to a network not similar to my VPN server. Thus, adding on top of the virtual machine, performance speed was relatively sluggish compared to the other machines and devices. While still being able to connect to the VPN server, I find that needing to connect any virtual machine to a VPN would be redundant since network would be based on what the hosting computer is connected to. In other words, if my Windows 10 computer is already running with the VPN, then it should automatically carry over to any virtual machine that I am operating on.



Figures 23: Android OS Test

Results and Drawbacks

With the experiments ending, I can say that configuring the VPN server was a success. Each machine and devices were able to connect to the Raspberry Pi server properly without any lag or connection issues. With a few errors and configuration changes, the Raspberry Pi server serves as a decent VPN hub that allows me to connect to my home network whenever I am away.

Comparing to third-party services, my cheaper alternative method was able to perform most of the functions that companies advertise their products to do. For instance, I was able to reroute to another network that is not associated with the public network.

Some features that my personal VPN server was not able to do however was having a geolocator function that allows me to access a VPN connection that is deemed to be located in another area. For instance, if I would like to access shows on Netflix that are exclusive to Japan, I will still not be able to access them as the shows are strictly region-locked, and that OpenVPN only applies to create a VPN server that directs to a single connection that you have configured it to be. That is definitely one downside, but only a con if the only reasoning for purchasing or making a VPN service was to access region-locked content.

Third-Party Services

I have mentioned that the secondary part of this project would be to compare the advantages and disadvantages of using either a personal VPN server or a VPN subscription.[12] Below is a simple comparison table made to distinguish both personal and subscription-based VPNs based on my findings. Both methods operates in a similar fashion of creating a tunnel straight to a trusted server. In the end, it truly is up to the user on what is deemed important for their own situation. If they prefer to use a VPN without the need of operating a server, then placing the trust on a company for providing the service is probably what they would want – as

long as the company stays true to the claims they have advertised. And for those that want their own full privacy and control, creating their own VPN service is available to them instead – as long as they have full or enough knowledge on starting one.

Personal VPN (Based on OpenVPN)	Third-Party Services
Pros: • Free/cheap alternative; available for every user to start on most devices • Keep hidden from ISP – but only if server is located externally from home • Full access to server – if set up in a familiar setting (home, work)	Pros: • Available for supported devices; high availability for users to purchase • Geolocation for many areas • Keep hidden from ISP – due to different server location
Cons: • No geolocation; strictly for personal/single access • Less features compared to alternative free VPNs (WireGuard)	Cons: • Subscription models/not free • Services may differ from each other • No full access to server; operates in different locations • Advertising as only security tool seen as misleading

Table 2: Personal VPN vs Third-Party VPN Comparison

CHAPTER 6

CONCLUSION

With the aid of the Raspberry Pi machine, contributors behind the PiVPN project source[10], and guides that helped me understand the concept[1][4][13], I would say that conducting this project was a success. Given the limited timeframe and state of the world, I was able to still showcase the ability of having your own personalized VPN server for a low price.

While I am able to operate my VPN server successfully, this does not mean that it comes with a few limitations and drawbacks. For instance, since my VPN server is connected to my personal home router, it will only work when I am away from my home network or when I am utilizing any external network connections. This means that I am not able to connect to the VPN server that I have made when connected to the same network that was used when configuring the service.

Despite its minor drawbacks, operating on the Pi was a fun learning experience and the VPN service is helpful whenever I need to connect to any public internet connections. Having a device or service that will help route me back to my home network is a big assistance in avoiding any potential man-in-the-middle attacks or allowing me to access internal sources on the go.

For future improvements, I would like to try out other open-source VPNs and see how they differ with OpenVPN (ie. Wireguard). Also, I wish to look over any potential security threats that the OpenVPN services faces and see how I can contribute to the project myself. As this is an open-source service, anyone can have the ability to try out OpenVPN themselves on supported devices. In a way, lending my service to them would be a nice gesture in thanking them for having the software readily available to everyone. And with that, my report concludes.

REFERENCES OR WORKS CITED

[1] C. Podszun, “How to Easily Configure a VPN on Raspberry Pi,” EXPERTE, 17-Feb-2021. [Online]. Available: https://www.experte.com/vpn/raspberry-pi.

[2] Chinmay, “How to SSH into a Raspberry Pi [Beginner's Tip],” It's FOSS, 13-May-2019. [Online]. Available: https://itsfoss.com/ssh-into-raspberry/.

[3] D. Gewirtz, “ExpressVPN vs. Surfshark vs. Nordvpn: Which is best?,” ZDNet, 27-Sep-2021. [Online]. Available: https://www.zdnet.com/article/expressvpn-vs-surfshark-vs-nordvpn-which-vpn-service-is-best/.

[4] Gus, “Build Your Own Raspberry Pi VPN Server,” Pi My Life Up, 11-Feb-2021. [Online]. Available: https://pimylifeup.com/raspberry-pi-vpn-server/.

[5] Kaspersky, “How a VPN can help hide your search history and is private browsing really secure?,” usa.kaspersky.com, 12-Jul-2021. [Online]. Available: https://usa.kaspersky.com/resource-center/definitions/how-does-vpn-keep-me-safe-online.

[6] M. Eddy, “The Fastest VPNs for 2021,” PCMAG, 15-Nov-2021. [Online]. Available: https://www.pcmag.com/picks/the-fastest-vpns.

[7] OpenVPN, “Hardening OpenVPN Security,” OpenVPN. [Online]. Available: https://openvpn.net/community-resources/hardening-openvpn-security/.

[8] OpenVPN, “Recommendations to improve security after installation,” OpenVPN. [Online]. Available: https://openvpn.net/vpn-server-resources/recommendations-to-improve-security-after-installation/#hardening-the-web-server-cipher-suite-string.

[9] OpenVPN, “TLS Error: TLS key negotiation failed to occur within 60 seconds (check your network connectivity),” OpenVPN. [Online]. Available: https://openvpn.net/faq/tls-error-tls-key-negotiation-failed-to-occur-within-60-seconds-check-your-network-connectivity/.

[10] PiVPN, “The PiVPN project,” PIVPN. [Online]. Available: https://www.pivpn.io/.

[11] Security Through Education, “Scarcity,” Security Through Education. [Online]. Available: https://www.social-engineer.org/framework/influencing-others/influence-tactics/scarcity/.

[12] VPN Store, “Personal VPN vs. business VPN: Know the Difference,” VPN Store. [Online]. Available: https://vpnstore.com/personal-vpn-vs-business-vpn/.

[13] W. Gordon, “How to create a VPN server with Raspberry Pi,” PCMAG, 20-Feb-2020. [Online]. Available: https://www.pcmag.com/how-to/how-to-create-a-vpn-server-with-raspberry-pi.

APPENDIX A: PROJECT PROPOSAL

CYB590 Master Project

Project Proposal

(Fall 2021)

Project title Personal VPN Server using Raspberry Pi

Proposed by Brandon Mao

Abstract

Virtual Private Networks (VPNs) allows end-users to sustain a private network connection while communicating over the Internet, adding extra layers of security over on the network. Many users may have heard of the benefits and security of using a VPN solely from third-party services providing marketing strategies. Companies would often utilize social engineering tactics, such as intimidation, urgency, and scarcity levels, to ensure users that they are not safe on the Internet without using their service now and provide them with an open window to purchase it before their discount sale is over. While some services can work fine, we are still entrusting our VPN purchases to third-party companies where the likelihood of data breaches, single point of failures (SPOFs), and end of service life (EOSL) can happen at any moment. Throughout this project, I will primarily demonstrate how to create your very own personal VPN server by utilizing inexpensive equipment and procedures including the Raspberry Pi 4 circuit board and open-source software to. In addition, this report will also cover the history and commercial uses of VPNs, provide an in-depth comparison and analysis between other third-party services versus creating your own VPN server, determine which method works efficiently in terms of network security.

References

[1] C. Podszun, "How to Easily Configure a VPN on Raspberry Pi," *EXPERTE*, 17-Feb-2021. [Online]. Available: https://www.experte.com/vpn/raspberry-pi.

[2] Gus, "Build Your Own Raspberry Pi VPN Server," *Pi My Life Up*, 11-Feb-2021. [Online]. Available: https://pimylifeup.com/raspberry-pi-vpn-server/.

[3] Kaspersky, "How a VPN can help hide your search history and is private browsing really secure?," *usa.kaspersky.com*, 12-Jul-2021. [Online]. Available: https://usa.kaspersky.com/resource-center/definitions/how-does-vpn-keep-me-safe-online. [Accessed: 23-Sep-2021].

[4] VPN Store, "Personal VPN vs. business VPN: Know the difference - vpnstore," *VPN Store*. [Online]. Available: https://vpnstore.com/personal-vpn-vs-business-vpn/.

[5] W. Gordon, "How to create a VPN server with Raspberry Pi," *PCMAG*, 20-Feb-2020. [Online]. Available: https://www.pcmag.com/how-to/how-to-create-a-vpn-server-with-raspberry-pi.

Jianchao (Jack) Han	Brandon Mao	09/26/2021
Faculty advisor	*Signature*	*Date*

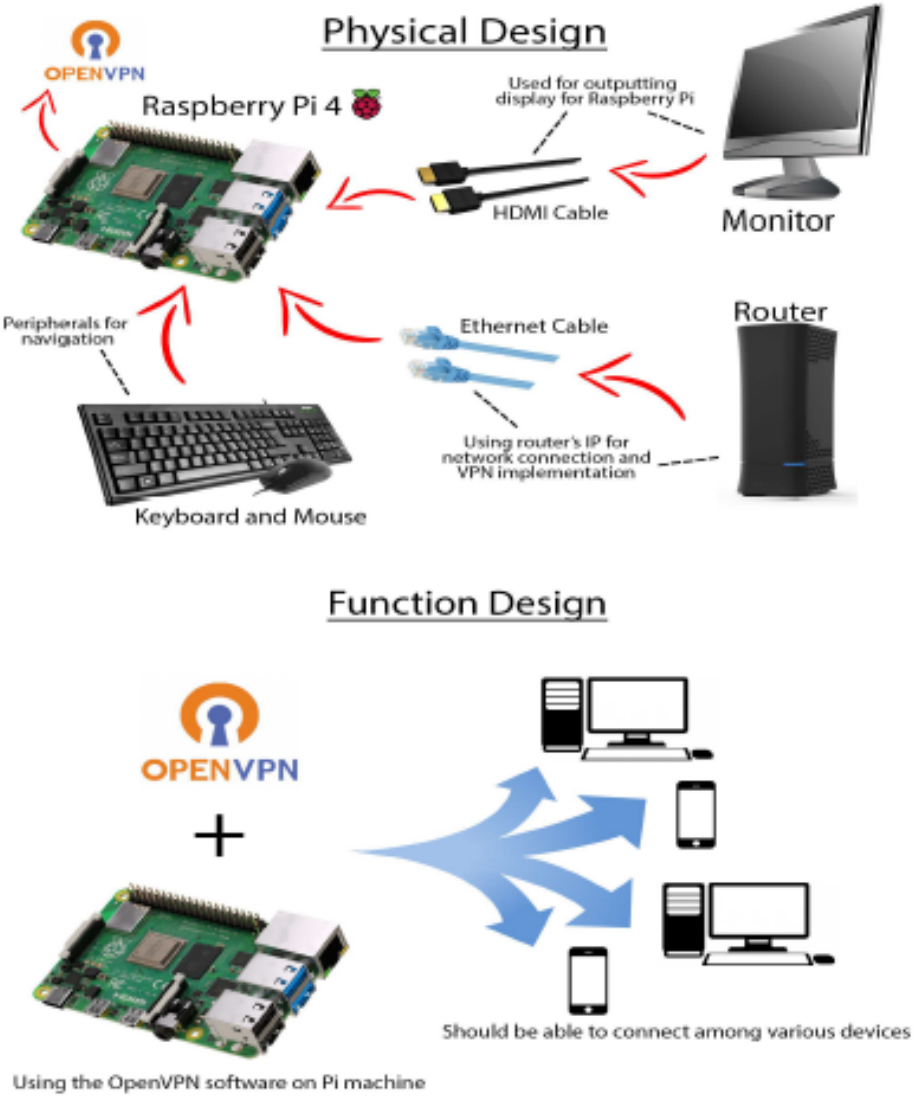
APPENDIX B: DESIGN REPORT

CYB590 – Master Project
Design Report
(Fall 2021)

Student Name: Brandon Mao	**Date: 10/17/2021**

Project Title: Personal VPN Server using Raspberry Pi

Project Design – Architecture/component diagram



Detailed functions of components:

- The project will consist of two aspects: the technical showcase where I intend to create the personal VPN server, and a comparison between the private server and a third-party service.
- The primary focus is on the VPN server that will take place on a Raspberry Pi 4 microcomputer being accessed with basic peripherals/KVM (keyboard, video, mouse). The default operating system installed is Linux, and will be configured by initiating Linux commands within the OS terminal. The Raspberry Pi computers have been used by many users as a safe platform to conduct various projects.
- OpenVPN will be the main software used to assist with the server (https://openvpn.net/). The software is open-source and available readily for free, with the commercial Access Server version coming at a price. OpenVPN is a virtual private network system allowing for secure point-to-point connections among various devices.
- After successful configuration of the personal VPN server, documentation on its capabilities, tests with multiple devices and machines, and any known errors and downsides will be reported and compared with the other public services.
- The second half of the project will focus on the comparative analysis of several third-party VPN services/subscriptions. Comparison will include popular services such as ExpressVPN, NordVPN, and SurfShark VPN. Discussion will include social engineering/marketing strategies, subscription prices, and capabilities with other devices and machines.

APPENDIX C: PROGRESS REPORTS

CYB590 – Master Project Progress Report
(Fall 2021)

Student Name: Brandon Mao **Date:** 10/03/2021

Project Title: Personal VPN Server using Raspberry Pi

Estimate of the project that has been done so far (percentage): 10%

Description of the project that has been done in this week (itemized):
From the proposal presentation, I have mentioned that I am going to spend the first half configuring the personal VPN server then work on the comparison research. At this time, I am setting up my Raspberry Pi with the necessary patches and updates before I go through with the VPN creation process. I am also currently working on the project design report and presentation. Within the design, I am still considering which software to use for the VPN server. So far, I am likely to utilize OpenVPN given its availability for open-source projects. I will make sure to address more on the design plans on the 17th.
Recap:

- Set up/update Raspberry Pi; Still deciding on VPN service
- Worked on Design Report & Presentation

Work to be done to complete the project (itemized) and plan to complete the project by the end of this semester:
Along with getting the VPN server to function, I would need to ensure proper management and usage within the next couple of weeks to document any findings for my comparative analysis with third-party solutions. In addition, research among third-party VPN providers will still need to be done and will be conducted within the halfway point of the semester. For integrity purposes, I do wish to try out a VPN subscription. Deciding on which service to use will still need to be considered.
Recap:

- Configure VPN server
- Comparison Analysis; Research Third-Party VPNs

CYB590 – Master Project Progress Report
(Fall 2021)

Student Name: Brandon Mao	**Date:** 10/03/2021

Project Title: Personal VPN Server using Raspberry Pi

Estimate of the project that has been done so far (percentage): 25%

Description of the project that has been done in this week (itemized):

During this time, I have decided to use OpenVPN, an open-source VPN service, to assist with the VPN server. So far, I am currently having my server being implemented and configured. I wanted to make sure to have an adequate understanding and research prior to operating on the server. As for the design presentation and report, I am wrapping it up and should be good to go before submission day.

Recap:

- Selected software (OpenVPN)
- Began configuration with Raspberry Pi; more research
- Working on Design Report and Presenation

Work to be done to complete the project (itemized) and plan to complete the project by the end of this semester:

With choosing the software completed and configuration almost wrapped up, I will need to focus onward with managing the server. This will include logging any errors that populate or any changes that have been modified during this research. Alongside, research among third-party services will still need to be done for my comparative analysis section of the project.

Recap:

- Finish up configuration; manage server
- Research third-party VPNs
- Work on comparison table
- Work on Final Project Report and Presentation

CYB590 – Master Project Progress Report
(Fall 2021)

Student Name: Brandon Mao	**Date:** 10/24/2021

Project Title: Personal VPN Server using Raspberry Pi

Estimate of the project that has been done so far (percentage): 50%

Description of the project that has been done in this week (itemized):
At this point of time, installation of the VPN server has been completed. I am having some issues configuring the VPN to my testing computer. With that said, daily management and documentation of its usage among machines and devices will be reported. This will include connecting to the VPN using computers and mobile phones from different areas. Also at this point, I will now begin researching various third-party VPN services to start on the comparative analysis part of the project. As mentioned previously, featured services may include ExpressVPN, SurfShark, and Nord.
Recap:

- VPN server set; need configuration
- Daily management and reporting

Work to be done to complete the project (itemized) and plan to complete the project by the end of this semester:

With the Project Design Presentation and Report completed, it is now time to start on the final documentation of the project. At this stage, I should have my VPN server up and currently logging any information or errors that have occurred. Along with that, other work to be done includes researching other VPN services and starting the comparison table.

Recap:

- Start on the Master Presentation and Report; set up for Progress meeting
- Research third-party services
- Begin conducting comparative analysis

CYB590 – Master Project Progress Report
(Fall 2021)

Student Name: Brandon Mao	**Date:** 10/31/2021

Project Title: Personal VPN Server using Raspberry Pi

Estimate of the project that has been done so far (percentage): 65%

Description of the project that has been done in this week (itemized):

After last week's configuration issue with setting up the client connectivity with the VPN server, I have successfully troubleshooted the problem. The dilemma of the personalized VPN server is that accessing the VPN server can only work outside of the home network since the server is using the home's router and IP address, meaning that testing a client on the same network would result in connection errors. I have tested the VPN service through my personal hotspot, LTE data, and external networks, with all being able to connect to the OpenVPN service. With the technical aspect out of the way, research with other VPN services will begin now.

Recap:

- Fixed configuration issue; VPN service now complete (technical part done)
- Research with third-party services commences

Work to be done to complete the project (itemized) and plan to complete the project by the end of this semester:

With the technical aspect out of the way, I will now start on the research side of the project. In short, the research side of the project will be to look over many of the third-party services that are offered on the Internet. A comparison will be part of the presentation where I plan to go over the features that are offered, the marketing strategies these companies use, and the pros and cons of using either a third-party service or personal server. In addition, the presentation for the Progress meeting is currently at work.

Recap:

- Research third-party services for comparative analysis
- Work on Progress Meeting presentation

CYB590 – Master Project Progress Report
(Fall 2021)

Student Name: Brandon Mao	**Date:** 11/14/2021

Project Title: Personal VPN Server using Raspberry Pi

Estimate of the project that has been done so far (percentage): 80%

Description of the project that has been done in this week (itemized):
With the server already set, I plan to research more into VPNs in order to develop the comparison table. So far, I am already looking over popular third-party services including ExpressVPN, SurfShark VPN, and NordVPN. I am specifically targeting these companies as they are the most advertised and sponsored VPNs that are currently prevalent around the media. The factors I wish to cover revolves around the marketing strategies and effectiveness that apply to their product, their choice of words and any inconsistencies, and how they display themselves to the public.

Recap:
- Continue researching on VPNs; how third-parties advertise
- Begin conducting comparison table

Work to be done to complete the project (itemized) and plan to complete the project by the end of this semester:

In addition to the research, I wish to look over any other hardening and security strategies that can help benefit with my current Pi VPN server. I seek to have this type of information available for my colleagues to have them understand various ways on how they can keep their server secured if they choose to create a project similar to mine. Work for the final presentation and report will also be conducted.

Recap:
- Provide additional security postures for server
- Work on Final Presentation and Report

CYB590 – Master Project Progress Report
(Fall 2021)

Student Name: Brandon Mao	**Date:** 11/21/2021

Project Title: Personal VPN Server using Raspberry Pi

Estimate of the project that has been done so far (percentage): 95%

Description of the project that has been done in this week (itemized):
At this time, both the technical aspect and research side of my project should be close to being completed. Until the date approaches for the Final Presentation, I will continue managing and documenting any changes, errors, tests, and status of the VPN server. Security changes will also be acknowledged and documented for the report and presentation slides.

Recap:

- VPN server set up; managing VPN server
- Research completed; proofreading for any grammar errors
- Finalizing report and slides

Work to be done to complete the project (itemized) and plan to complete the project by the end of this semester:
Most of the work that needs to be done revolves around finalizing all of the data and information on my report and presentation slides before submission/ presentation day. This includes checking for any errors and practicing for my presentation. With the given scope and circumstances, it seems that I was able to get a majority of my proposed project completed and looks to function successfully.

Recap:

- Last minute checkups with VPN server
- Finalize report and presentation before submission
- Practice presenting for Final Meeting