Fall 2020

MySmartPi

Camila Pulz de Faria

Fernando Aires da Silva

Jesus Colina Nunez

Reginaldo Pereira dos Santos

Thenilde Borges

cct | College Dublin
Computing • IT • Business

MySmartPi

Applied Technology Group Project

Supervisor: Greg South

Camila Pulz de Faria 2017162

Fernando Aires da Silva 2017243

Jesus Colina Nunez 2017156

Reginaldo Pereira dos Santos 2017202

Thenilde Borges 2017161

Table of Contents

ABSTRACT

Nowadays, accessing the Internet in a secure way in a big concern for many people due to the increase of cybersecurity attacks and the vulnerability of the data that is transferred online. In order to address such vulnerabilities, the use of a Virtual Private Network is really important. Not only for security reasons, but also to access resources of the network, such as printers, files or web pages. Considering that many people, especially IT students, have curiosity and enjoy creating their own technologies, this project aims to create a user manual to teach how people can create their own VPN server at home using a Raspberry Pi and to access their files and folders which are in the network. For that, tutorials were used and adapted in order to install the VPN server and NAS. In order to prove that the whole process was successful, some tools, such as, Wireshark, were used to show how the network traffic works once the VPN is used. The process was successful and many concepts were learnt and used such as Cryptography, Port forwarding, dynamic DNS, OpenVPN, etc.

Key words: VPN, OpenVPN, NAS, Cryptography, port forwarding, dynamic DNS

ACKNOWLEDGMENTS

The group smart5 wish to sincere appreciate the help and support from the supervisor Greg South since the beginning of this project who encourages us to keep doing a good project and helping understand the whole process. For all his online sessions during this pandemic time which were so important for the development of this project.

We wish to acknowledge the support of our family, partners and close friends who were always there to listen and give support in moments of pressure and hopeless. Especially, my partner Aris Vakondios who stayed by my side for the whole year giving me support when I most needed it. Also, the members of the smart5 team that in many times were not only my colleagues, but friends who help me when I was feeling desperate.

Introduction

Nowadays, using the internet in a secure and private way has become a big concern for many users due to the increasing amount of cybersecurity attacks that has been happening on a daily basis against public and non-public organizations, government and individuals (Janson and Bruijn, 2017).

The hack of the Sony Pictures occurred in 2014, as an example, on which hackers leaked some of the unreleased films and scripts online, sensitive information such as emails, passwords and salaries of the executives and employees (Alvarez, 2014), confirms the need for cybersecurity measures to be discuss and placed to prevent and deal with such threats in the future.

This project aims to present a solution to protect the user and its data while using the Internet and an alternative to remotely access files and folders stored in their home network. For that, the contents are divided into six chapters and appendices.

Chapter 1 it is the introduction of section and shows the problem to be solved and objectives and goals. Also, introduces important definitions and explanations about the technologies that are going to be use, for example, VPN, types and protocols, Cryptography, etc.

Chapter 2 discusses the System Analyses with the usage of diagrams that explain how the connections through VPN should be in order to achieve the main goals, activity diagram, etc.

Chapter 3 shows how the system will look like. For that, some definitions were important such as port forwarding, dynamic DNS, OpenVPN which are important to understand what it is needed for the implementation.

Chapter 4 illustrates the implementation of the proposed project. Showing and explaining step by step how to install the VPN server and the NAS feature in the Raspberry Pi. Also, it describes how to configure the router to enable port forwarding and how to use the dynamic DNS service.

Chapter 5 analyses and tests the installation of the VPN and NAS showing via screenshots the results of the process and how their usage is verified with the use of tools, such as Wireshark, websites that verify IP address.

Chapter 6 discuss the process of elaboration of this project reflecting on the main objectives and goals of the project and its implementation and achievements.

Chapter 1

This chapter introduces the idea of this project by discussing the problem which the technology intends to solve. Also, defines important concepts about VPN, Cryptography, Wireshark and Linux that will be explored in the next chapters.

Problem

The awareness of cybersecurity is not only essential for big organizations or the government, but as well for individuals who use the internet to access their social media accounts, online services or webpages. Sharing data over the Internet introduces a vulnerability that can be exploited easily by hackers in order to blackmail someone or use their information for malicious purposes (Janson and Bruijn, 2017).

Using the public Wi-Fi in coffee places, for example, increases even more the vulnerability of the data transmitted since it is a public network and there are no guarantees about the equipment provided by the coffee place; therefore, another person could connect to the same Wi-Fi hotspot to monitor the packets sent in the network, stealing passwords and seeing the webpages and service accessed by the user (Android Authority, 2018).

A public network can be defined as a large collection of unrelated peers that exchange information relatively freely with one another while the private network is composed of computers owned by a single organization that share information only with one another (Scott, Wolfe and Erwin, 1999).

On this context, what the user can do to secure themselves and their data while browsing the Internet? Moreover, how people can access a public network safely as if they were in their private home network? More specifically, those people who are travelling and need to connect to a public network in their hotels or coffee shops and wish to access a specific file or folder which is stored in their home network?

Solution

The solution for the problem is the use of Virtual Private Networks to create a secure connection between the devices of the user and their home network allowing them to access their private network and its resources remotely from anywhere in a safe way.

The use of the VPN not only could improve the security of the network traffic, but it also allows the users to access contents and webpages from other countries (Android Authority, 2018). As an example, on a trip to China where websites and services, such as Facebook, are blocked, the users could create a connection through VPN to access the contents in their home countries.

Furthermore, the use of a VPN allows the user to access files, music, movies from anywhere, access a home printer remotely, bypass firewalls and websites restrictions, connect and control home cameras and smart devices over the network (Phillips, 2018).

Although Virtual Private Networks are a great solution for concerns regards to privacy and security, there are some disadvantages of its use. The speed of the connection is slower comparing to connections without VPN, also the VPN server can become overloaded and the service be interrupted. (Android Authority, 2018).

Another disadvantage is the fact that the users need to be careful when choosing a VPN server provider, because those companies hold their information and could use it in a malicious way. For instance, some companies operate in countries that also have some security restrictions and, to continue operate their business, they have to play with both sides (Eddy, 2019).

Considering the disadvantages mentioned before, the users can create their own Virtual Private Network by using a Raspberry Pi as an alternative. By doing so, they could feel safe about their information and learn how to create their own device instead of relying on third parties. Especially IT students who are curious about learning how network system works and enjoy the learning process of creating their own technologies.

Goals

This project aims to install and run a VPN server on a Raspberry Pi to offer a secure way of connecting through the VPN to a home network in order to access remotely the files and folders in the network. Moreover:

- Discuss the importance of taking network security measures while using public networks.
- Discuss the technologies and protocols that are necessary for the implementation of a VPN and NAS.
- Create a user manual to be used for educational purposes to teach IT students how to create their own Virtual Private Network and NAS.

Objectives

- Execute the installation of OpenVPN in a Raspberry Pi using Raspbian operating system.
- Connect the client devices to the OpenVPN server by using the client certificate.
- Demonstrate the usage of the Virtual Private Network by using Wireshark and other resources, such as websites and commands.
- Install software to allow Network-Attached Storage in the home network.
- Demonstrate that the folders and files can be accessed from a device located in a different network by using the VPN connection.

Virtual Private Networks

The history of VPNs starts in 1996 when the Peer-to-Peer Tunnelling protocol (PPTP) was developed by Microsoft to create a more secure and private connection between a computer and the Internet (Mocan 2018).

Originally, Virtual Private Networks were only used by business and governmental organizations to protect the data and avoid that it could be exposed in the Internet, and, nowadays, more powerful encryption standards are implemented to guarantee the security of the data (Mocan, 2018).

The VPN uses tunnelling protocols that it sets up secure connections that hide the source using high-level encryption which allows the data to go over an encrypted connection from one device to another point of the Internet (Android Authority, 2018).

The VPN permits users to create a secure, private network over a public network such as the Internet. They can be created using software, hardware or a combination of both which creates a secure link between peers over a public network (Scott, Wolfe and Erwin, 1999).

By using VPN, it is also possible to shield the IP address and the location of the users, because once the network traffic goes through a VPN the IP address is the VPN server provider, not the original IP address of the users. (Android Authority, 2018).

Types of VPNs

Remote- Access VPN

The remote-access VPN, also known as VPDNs (virtual private dial-up networks), allows the user to connect to a private network and access its system and resources remotely by connecting through Internet in a secure and private way (VPN One Click, 2019).

It is normally used in business organizations to make possible that employees connect remotely via a private LAN connection to the network of the organization. The VPN service is usually outsourced by an Email Service Provider (ESP) or a service provider and a NAS (Network-Attached Storage) is configured to provide the users the access to the resources (Tyson, Pollette and Crawford, 2020).

Site-to-site VPN Access

The site-to-site VPN, also called router-to-router VPN, is utilized by organizations with offices in different geographical locations to securely connect the networks of one another (VPN One Click, 2019). Therefore, this type of VPN expands the network of the companies by allowing the access to their network by their employees who are in different locations (Tyson, Pollette and Crawford, 2020).

It is called router-to-router due to the fact that during the communication, one router is the VPN Client meanwhile the other router is the VPN Server. In addition, both routers are authenticated (VPN One Click, 2019).

Protocols

The Virtual Private Network is the result of virtual networking connections combined with cryptographic algorithms. Its security is based on the effectiveness of its authentication and encryption protocols, and both authentication and encryption are required to keep the virtual network traffic protected and private. (Schneier, n.d.).

VPN protocols defines exactly how the data routes between the computer and the VPN server. They have different purposes and benefit the users in a range of circumstances. For example, some of them prioritize speed, while others are dedicated to privacy and security (Phillips, 2017).

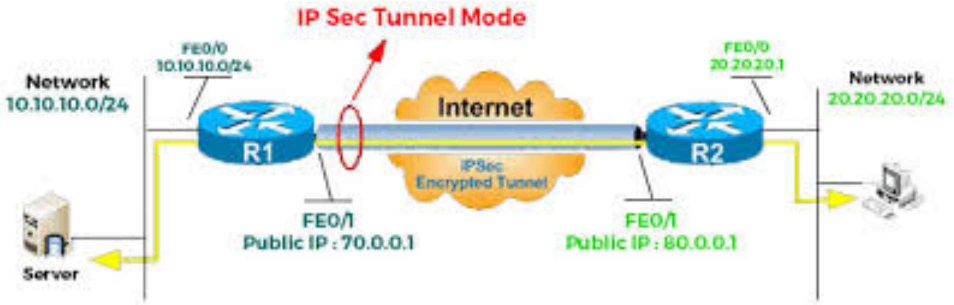
According to Phillips (2017) those are the most common VPN protocols.

- OpenVPN: Open source, offers strongest encryption, suitable for all activities, if a little slow at times
- L2TP/IPsec: Widely used protocol, good speeds, but easily blocked due to reliance on single port
- SSTP: Good security, difficult to block and detect.
- IKEv2: Fast, mobile friendly, with several open source implementations (potentially undermined by NSA)
- PPTP: Fast, widely supported, but full of security holes, only use for streaming and basic web browsing

IPsec Protocol

The Internet Protocol Security does the authentication of the session and encrypt each packet data during the connection. It operates in Transport mode encrypting the message of the packets and Tunnelling mode encrypting the entire data packet (VPN One Click, 2019).

The most common used of the IPsec Protocol it is commonly used in site-to-site or device-to-device VPN.



Layer 2 Tunnelling Protocol

Layer 2 Tunnelling Protocol (L2TP) is a tunnelling protocol that it is usually combine with another protocol to create a safer and higher secure connection. It creates a tunnel between two L2TP connections points while IPsec encrypts the data (VPN One Click, 2019).

According to IBM Knowledge Centre (2020) the Layer Two Tunnelling Protocol (L2TP) separates the locations at which the dial-up protocol ends and where the access to the network is provided (Phillips, 2017).

Layer Two Tunnelling Protocol (L2TP) is an extension of the Point-to-Point Tunnelling Protocol (PPTP) used by an internet service provider (ISP) to enable the operation of a virtual private network (VPN) over the internet (Rouse, 2007).

L2TP is the Layer Two Tunnelling Protocol, an extension of PPTP, which combines the latter with L2F (Layer 2 Forwarding Protocol) that was designed by Cisco. L2TP does not have integrated encryption, so this feature is added by IPsec (Internet Protocol Security) (DeMuro, 2020).

- Voluntary tunnel is created by the user;
- Compulsory tunnel model – incoming call is a tunnel created without any action from the users;
- Compulsory tunnel model – remote dial is the home gateway initiates a tunnel to an internet service provider;
- L2TP multi-hop connection used to redirecting L2TP traffic on behalf of client L2TP access concentrators and networks servers.

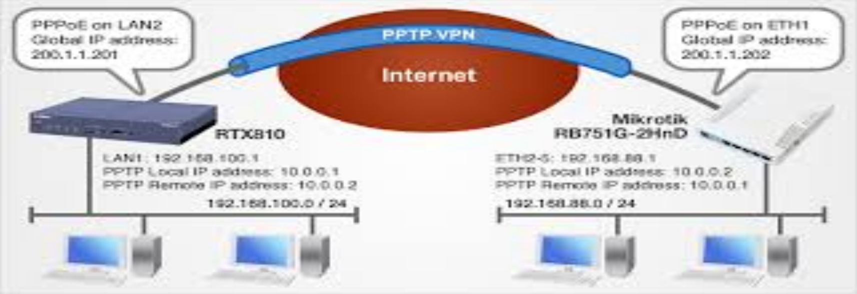
Point-to-point Tunnelling Protocol (PPTP)

The point-to-point tunnelling protocol (PPTP) was created by a group of engineers from different organizations to create a virtual private network between remote access users and network servers (Crist and Keijser, 2015).

The PPTP is an extension of the Point-to-Point Protocol which does not provide traffic encryption. For that, it uses PPP to encrypt traffic from different protocols and encapsulate the PPP frames in the IP datagrams to be sent across an IP network through a TCP connection (Crist and Keijser, 2015).

Breaking into a VPN is often the same as penetrating the firewall. The Point-to-Point Tunnelling Protocol (PPTP) was designed to solve this problem of creating and maintaining a VPN over a public TCP/IP network using the common Point-to-Point Protocol (PPP) (Schneier, n.d.).

Point-to-Point Tunnelling Protocol is one of the oldest VPN protocols, but it still in use in some places. However, the majority of services have long upgraded to faster and more secure protocols. (Phillips, 2017).



Secure Sockets Layer (SSL)

It was originally developed by Netscape in 1994 and it became very popular for transferring data securely between servers and applications across the Internet. It provides message authentication, confidentially and integrity by using a combination of underlying cryptographic protocols. In the OSI model, SSL acts between the application and Transport Layers (Phillips, 2017).

Secure Sockets Layer (SSL) and Transport Layer Security (TLS) create a VPN connection on which the web browser works as the client and the access is restricted only to specific applications (VPN One Click, 2019).

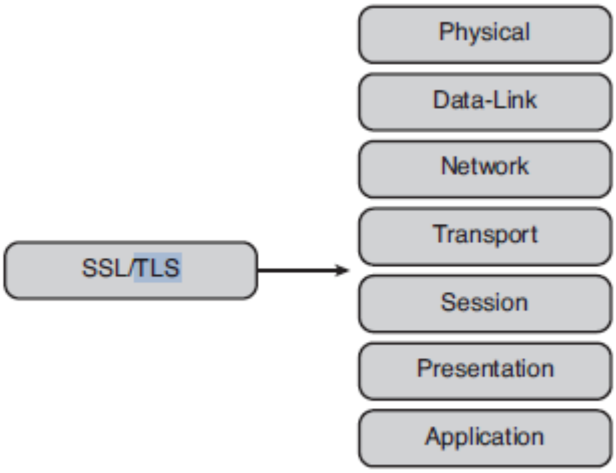


Figure 9-1 *SSL's OSI Layer Position*

Transport Layer Security (TLS)

The Transport Layer Security (TLS) is a popular security protocol that provides privacy and data security for communications over the Internet by ensuring that the devices who are trying to communicate are in fact who they claim they are by authentication mechanisms, providing data integrity and confidentiality through encryption. (LAKE, 2019).

The most common use of TLS protocol is the encryption of the communication between web applications and servers, such as web browsers loading a website (CloudFlare, 2020).

According to Lake (2019), one of the most important functionalities of TLS it is the record protocol which is the underlying protocol responsible for the structure of the process.

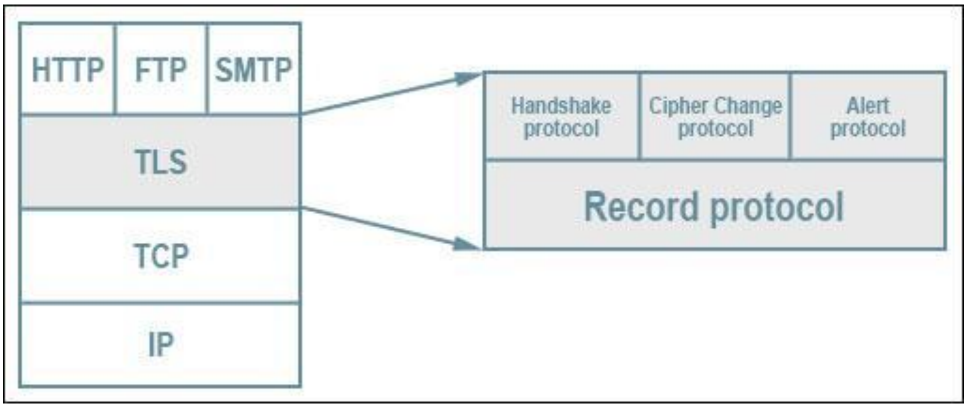


Diagram showing the TLS stack. ***TLS protocol stack*** *by Jeffreytedjosukmono. Licensed under CC0.*

The record protocol contains five separate subprotocols formatted as records (LAKES, 2019).

- **Handshake** – this protocol is used to set up the parameters for a secure connection.
- **Application** – the application protocol begins after the handshake process, and it is where data is securely transmitted between the two parties.
- **Alert** – the alert protocol is used by either party in a connection to notify the other if there are any errors, stability issues or a potential compromise.
- **Change Cipher Spec** – this protocol is used by the client or the server to alter the encryption parameters.
- **Heartbeat** – This is a TLS extension that lets one side of the connection know whether its peer is still alive, and prevents firewalls from closing inactive connections.

The most important ones to understand for this project are the handshake and the application protocols, because these are responsible for establishing the connection and then securely transmitting the data. (LAKES, 2019).

The steps for the TLS handshake might change depending on the kind of key exchange algorithm used and the cipher that is supported by both sides. However, the RSA key exchange algorithm is the most frequently used. The process goes as described (TLS Handshake Learning Objectives, 2020).

- The 'client hello' message: The client initiates the handshake by sending a "hello" message to the server. The message will include which TLS version the client supports, the cipher suites supported, and a string of random bytes known as the "client random."
- The 'server hello' message: In reply to the client hello message, the server sends a message containing the server's SSL certificate, the server's chosen cipher suite, and the "server random," another random string of bytes that's generated by the server.
- Authentication: The client verifies the server's SSL certificate with the certificate authority that issued it. This confirms that the server is who it says it is, and that the client is interacting with the actual owner of the domain.
- The premaster secret: The client sends one more random string of bytes, the "premaster secret." The premaster secret is encrypted with the public key and can only be

decrypted with the private key by the server. (The client gets the public key from the server's SSL certificate).

- Private key used: The server decrypts the premaster secret.
- Session keys created: Both client and server generate session keys from the client random, the server random, and the premaster secret. They should arrive at the same results.
- Client is ready: The client sends a "finished" message that is encrypted with a session key.
- Server is ready: The server sends a "finished" message encrypted with a session key.
- Secure symmetric encryption achieved: The handshake is completed, and communication continues using the session keys.

All the TLS handshakes make use of asymmetric encryption: the public and private key (TLS Handshake Learning Objectives, 2020). Once a handshake has occurred, TLS allows uses a shorter handshake process. These handshakes use the session ID to link the new connection to the previous parameters. (LAKES, 2019).

OpenVPN

OpenVPN is one of the most important protocols that is open source and is one of the protocols that offers more security among the others. It allows the users to protect their data by essentially using a combination of unbreakable AES-256-bit key encryption (amongst others), with 2048-bit RSA authentication and a 160-bit SHA1 hash algorithm. (Phillips, 2017).

OpenVPN provides secure authentication for the users by using a public (static) key, corresponding certificate or user name and password. Moreover, its use also includes membership in a broad user community that provides supports, suggestions and/or additions of new functionalities (Skendzic and Kovaci, 2017).

OpenVPN is available for both protocols OpenVPN TCP and OpenVPN UDP. Some VPN providers do not give the option of choosing between one or another, while some offer both to the users, but not much information about what is the best option and the difference between them (DeMuro, 2020).

Secure Shell (SSH)

Secure Shell is a protocol that establishes how devices can securely communicate over a network. by providing alternatives for strong authentications and encryption to guarantee the security and the integrity of the communication (Ssh.com, 2013).

The SSH client initialises the communication by using public key cryptography to verify the identity of the server and, then, the SSH protocol uses strong symmetric encryption and hashing algorithms to guarantee the integrity of the data being transferred between client and server (Ssh.com, 2013). In order words, Secure Shell (SSH) creates the VPN tunnel for the data to be transferred and it ensures the tunnel is encrypted (VPN One Click, 2019).

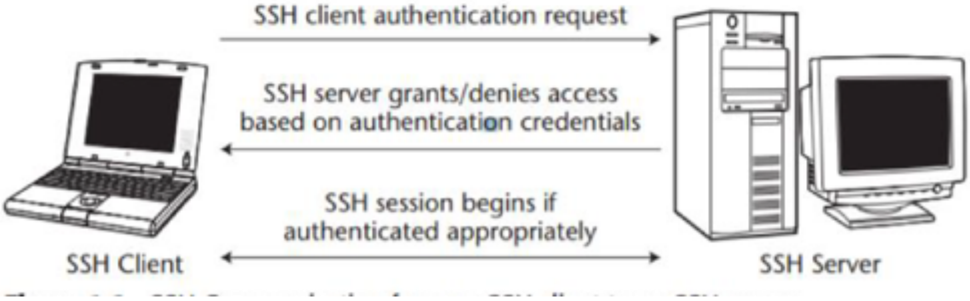

SSH client authentication request
SSH server grants/denies access based on authentication credentials
SSH session begins if authenticated appropriately
SSH Client
SSH Server

Diagram (Ssh.com. (2013).

Examples of VPN server providers

Many VPN service providers are on the market today and it can be a hard task choosing one. According to the website *Tom's guide*, ExpressVPN occupies the first position on the rank list of the best VPN of 2019 in terms of fastest speed, number of connections, customer service support and wide compatibility of devices, such as Mac, Windows, IOS, etc. (Rivington, 2019).

According to the ExpressVPN website (2019), their VPN offers access to contents from anywhere, over hundred server locations, anonymous browsing, IP address masking as basic features.

The advanced features provide the best in class encryption, trusted server technology – which means no data is written into the hard drives – no activity or connection logs (traffic data, DNS queries, etc.). Moreover, it offers split tunnelling to let the user route some device traffic through a VPN while the rest access the internet directly. For the last, it, if the VPN connection fails it blocks all the internet traffic until protection is restored (ExpressVPN, 2019).

Although, there are efficient VPN service providers in the market, some users prefer to set their own VPN at home. The reasons lie is the fact that some people dislike the idea of having to rely on someone else to protect their privacy and for curiosity about how VPN servers work. In addition to it, some small business organizations choose to setting up their own VPN server in the office to secure the remote access to the company network (Athow, 2017).

Thinking about the curiosity aspect, it is important to include students of Information Technology that are motivated to understand networks and willing to create their own technologies in order to understand the concepts and improve their learning process by doing practical projects.

Raspberry Pi

The Raspberry is a low-cost computer that has the size of a credit card which use it is possible by just plugging a standard keyboard and mouse to it and connect to a monitor or television (Raspberry Pi Foundation, 2019).

The Raspberry Pi was created by Eben Upton, Rob Mullins, Jack Lang and Alan Mycroft, who were members of the University of Cambridge Computer Lab in UK, to teach computing and programming to young people, in 2006 (Raspberry Pi Foundation, 2019).

Between 2006 and 2008, the foundation started to develop several prototypes based on the Atmel ATmega644 microcontroller, which would serve the structure for the Raspberry Pi (Raspberry Pi Foundation, 2019). Nowadays, it is very popular cheap, practical and affordable alternative for people of different ages to explore a variety of capabilities of computing. It also facilitates programming learning in languages such as Scratch and Python (Raspberry Pi Foundation, 2019).

Linux

Linux is one of the most important technologies inventions of the 21st century, not only due its impact in computing systems and Internet, but because it shows how collaborative projects achieve greater and faster results than the ones done by single individuals or specific companies (Negus, 2015).

Linux is an operating system that consists of the software that manages the computer and runs applications on it, like any other operating system. According to Negus (2015), as any other operating system, Linux does the actions of detecting and preparing hardware once the system boots up, keep track of the processes running at the same time and manage which and when they can access the CPU. It also manages memory, provides the user graphic interface, controls ownership and access to the files and directories in the filesystem, provide users access and authentication services, offer administrative utilities, starting up services and programming tools (Negus, 2015).

The fact that Linux is open source and allows the users to make changes to the source code, to debug or run their own software, made Linux so popular among individuals and multi-billion-dollar companies, such as Google, Facebook, etc. (Negus, 2015).

Linux Torvalds created Linux as a hobby in 1992 while he was a student at the University of Helsinki, Finland. He created it not to meet market demands, but by a desire to produce a new technology that could overcome the barriers to produce programs back in the time (Negus, 2015).

The Linux distribution is needed to use Linux. The Linux Distribution includes all the standards components of the Linux system, also a set of administrative tools to facilitate the installation and upgrade process of the operating system (Silberschatz, Galvin and Gagne, 2014).

Wireshark

Wireshark is an open source protocol analyser created by Geraldo Combs that runs on many computers that use Windows or UNIX Platforms. Its original name was Ethereal and nowadays it is widely used to analyse network traffic and the communication between networks (Vanparia, Ghodasara and Donga, 2003).

It is used on many numbers of computers without concerns about license keys or fees and it can be used to troubleshoot network problems, network security problems, verify network applications, debug protocols implementations, and learn about internals network protocols (Sharpe, 2019).

Wireshark provides a range of filters that help to refine the search criteria according to the number of the protocol. It supports more than 1000 protocols which can be analysed by layers and it is the best tool to analyse and verify that the VPN is encrypting the data (Vanparia, Ghodasara and Donga, 2003).

Some of its features include live capture of packets data from a network interface, filter packets based on different criteria, display packets with their protocol information, create statistics, etc (Sharpe, 2019).

Cryptography

Cryptography is the area of study which analyses different schemas and techniques used for converting **plaintext** messages format into a coded message called **cyphertext** using algorithms**.** This process is called **encryption,** while the opposite process of converting the coded message into its original format is called **decryption** (Stallings, 2007).

The cryptographic algorithms are classified into three categories, such as: symmetric algorithm which is normally used for privacy and confidentiality; asymmetric algorithm which is usually used for authentication, non-repudiation and key exchange; and hash functions that are normally used for message integrity (Kesler, 2020).

Symmetric Encryption Algorithm

This algorithm is named symmetric or secret-key algorithm because both sender and receiver share the same secret or single key. The sender uses the key to encrypt the message generating and sending the cyphertext message and the receiver uses the same key to decrypt the cyphertext into the original message (Kessler, 2020).

The symmetric encryption algorithms operate by either using substitution, on which the elements of the plain text are replaced by another element; or using transposition which the elements in the plain text are rearranged (Stallings, 2007).

The algorithm can process the input by blocks of elements at each time producing an output block for each input block (block cypher) or the elements are continuously processed producing the elements one at time as the processes happens (stream cypher).

The most common algorithm used for symmetric encryption is block cyphers (Stallings, 2007). The plaintext input is processed in fixed-size blocks and the output is a cyphertext of equal size of each plaintext block (Stallings, 2007).

Data Encryption Standard (DES)

This is of the famous encryption scheme that it is based on the Data Encryption Algorithm (DEA). The plaintext has 64 bits of length and the secret key has 56 bits. The process occurs within 16 rounds on which 16 subkeys are generated from the originally 56 bits key and used one for each round (Stallings, 2003).

Triple DES (3DES)

The algorithm uses three keys and three executions of the algorithm DES. It has a key length of 168 bits which makes the encryption stronger and more resistance to brute force attacks (Stallings, 2003). Since it is a symmetric algorithm, it uses only one key for encryption and decryption.

According to Stalling (2007), the function can be represented by:

$$C = E_{K3}[D_{K2}[E_{K1}[P]]]$$

Where C= ciphertext;

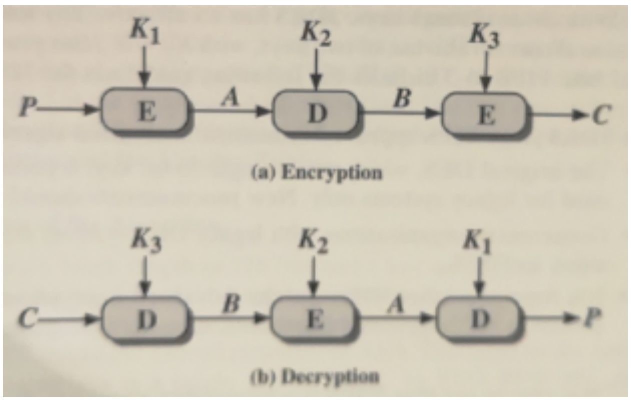
P=plaintext

$E_K[X]$ = encryption of X using key K

$D_K[Y]$ = decryption of Y using key K

While the decryption is the same function, but with reverse keys.

$$P = D_{K1}[E_{K2}[D_{K3}[C]]]$$



Stallings, 2003, p. 37

Asymmetric Encryption Algorithm

Asymmetric Encryption uses the idea of dual key on which the public key is the one that allows the encryption and the private key is used for decryption. Both keys can be used for encryption while the other is used for decryption (Kesler, 2020).

Asymmetric Encryption Algorithm is not based on substitution and permutation algorithms, but, instead, it relies on mathematical functions. The public key and the private key complement each other to perform operations, such as encryption and decryption or signature generation and signature verification (Stallings, 2007).

Some important concepts about asymmetric encryption are public key certificate and public key infrastructure (PKI). The first term refers to a digital document created and signed by a private key of a Certification Authority that maps the name of a subscriber to a public key. The certificate shows that the subscriber identified in the certificate has control of the private key. Meanwhile, PKI is a set of policies, processes, server platforms and workstations that work together to administrate, create, maintain and revoke certificates and public keys (Kesler, 2020).

Although the asymmetric encryption uses two keys while symmetric uses only one secret key, it is not more secure than the last one. The security of encryption depends on the length of the key and the amount of work to break a cypher (Stalling, 2007).

According to Stalling (2007), the public-key scheme has six components:

- plaintext: the readable message that is going to be encrypted.
- encryption algorithm: the algorithm that will transform the plaintext into a cypher text.
- public and private keys: the pair of keys that are selected for the process. If one key is used for encryption the other one will be used for decryption.
- cyphertext: it is the result of the encryption algorithm.
- decryption algorithm: the algorithm transforms the cyphertext into plaintext

RSA

The Rivest-Shamir-Adleman (RSA) it is the most important encryption/decryption algorithm in use and it was created by Ron Rivest, Adi Shamir and Len Adleman in 1977 (Stalling, 2007). It is a combination of two algorithms: key generation to produce the public and private keys and RSA function evaluation to take handle encryption and decryption processes (Farik and Nisha, 2015).

The RSA algorithm has become a popular method for encryption because it allows both of the keys to be used to encrypt the message while the other does the opposite by decrypting the message (Farik and Nisha, 2015).

The key-pair is created from a very large number **n** which is the product of two prime numbers which has to be 100 or more digits in length to generate a number n with double digits as the original prime numbers. Also, the public key contains **n** and a derivative of one of the factors of **n** (Kesler, 2020).

Some authors defend that the effectiveness of the RSA algorithm is due to the complexity of factoring large integers into prime numbers. (Farik and Nisha, 2015). However, others do not agree with that and state that algorithm is secure because the attacker cannot find the value of the prime factors of **n,** therefore, the private key (Kesler, 2020).

According to Kesler (2020) the following steps should be taken to create an RSA public and private key:

- Choose two prime numbers (**p** and **q**) and calculate the modulus **n. (n= pq)**
- Choose a third number **e** that is prime to the product **(p-1)(q-1)**
- Calculate an integer **n** from the quotient **(ed-1)/[(p-1)(q-1)]**
- The public key is the value of **(n, e).**
- The encryption of the message, therefore, the creation of the cyphertext **C** can be represented by the function: $\mathbf{C = M^e \bmod n}$ while the decryption of **C** can be expressed by the equation $\mathbf{M = C^d \bmod n}$ (Kesler, 2020).

Diffie-Hellman Key Exchange

It is an algorithm created by Diffie and Hellman and it is one of the simplest asymmetric algorithms. Its purpose is to allow two users to securely exchange a key that can be used for symmetric encryption of the messages (Stalling, 2007).

The table below briefly explains how the algorithm works. At the end of the algorithm both users exchange a secret value.

User A	**User B**
User A and B share a prime number q and an integer a. Rules: a < q and a is a primitive root of q.	User A and B share a prime number q and an integer a. Rules: a < q and a is a primitive root of q.
User A generates a private key X_A, where $X_A < q$.	User B generates a private key X_B, where $X_B < q$.
User A calculates a public key based on the equation: $Y_A = a^{X_A} \bmod q$	User B calculates a public key based on the equation: $Y_B = a^{X_B} \bmod q$
User A receives the public key Y_B from user B in plaintext	User B receives the public key Y_A from user A in plaintext
User A calculates shared secret key based on the equation $K=(Y_B)^{X_A \bmod q}$	User B calculates shared secret key based on the equation $K=(Y_A)^{X_B \bmod q}$

(Based on Diagram from **Cryptography** - Stallings, 2007)

Hash Functions

Hash algorithms are the ones which do not use any key and it is normally used for message integrity. It contains hash values with fixed lengths based on the plaintext making impossible the recovery of the contents of the plaintext or its length (Kesler, 2020).

They are used to ensure the integrity of the data through mechanisms that provide a digital fingerprint of the contents of the file to guarantee that the file was not altered by an attacker or virus (Kesler, 2020).

There are many hash algorithms, but for this project only the SHA is covered because it is related to the VPN implementation on chapter 4.

SHA

Secure Hash Algorithm are hash functions that processes a message to produce a truncate representation called *message digest* and preserving the integrity of the message: in case there is any change to the original message, the message digest will be different as well (Kesler, 2020).

The algorithm works in two way: pre-processing and hash computation. The pre-processing consists in encapsulation the message, converting it into m-bits blocks and initializing values to be used in the hash function. Meanwhile, the hash computation generates a message schedule from the encapsulated message and processes it by using functions, word processes, constants, etc, to create a set of hash values. Lastly, the final hash value is used to determine the message digest (Stallings, 2003).

The algorithms are SHA-1, SHA-224, SHA-256, SHA-384, SHA-512/224, SHA512-256 and they differ according to the size of the blocks and the words of the data that are used during the hashing or which compound the message digest (Stallings, 2007).

The size of the message block **m-bits** depends on the algorithm:

SHA-1, SHA-256, SHA-224: each message block has 512 bits which corresponds to a sequence of sixteen **32-bit words.**

SHA-384, SHA-512/224, SHA512-256: each message block has 1024 bits that correspond to a sequence of sixteen **64-bit words**.

Network-Attached Storage

With the popularity of household networks, the NAS (Network-Attached Stored) are starting to become popular in both residences and small and medium-sizes enterprises for bringing a fairly simple configurations compared to servers and at least at their cost. The basic functions of NAS most be storage services to all network connected computers.

Network-Attached Storage (NAS) is a dedicated file storage that enables multiple users and different client devices to retrieve data from the same centralized disk capacity (TechTarget, 2019). Users on a local area network (LAN) access the shared storage over the network. The NAS connects to a wireless router, which facilitates for distributed work environments to access files and folders from any device connected to the network (TechTarget, 2019).

It is a type of file storage device connected to a computer network that usually contains several drives designed to hold databases, system images, and backup files in a central location. The stored data is always available to multiple users to access it simultaneously (Flack, 2020).

Due to the fact that NAS runs independently of the network server, it allows users to access data without interruption even if the server goes down. Besides, it increases the speed of file sharing since it is only responsible for file storage and retrieval (Flack, 2020).

There are two main types of networked storage: NAS and storage area networks (SANs). While NAS handles unstructured data, such as audio, video, websites, text files and Microsoft Office documents, SANs are designed primarily for block storage inside databases, also known as *structured data* (SearchStorage, 2019).

In order to enable Network-Attached Storage in the Raspberry Pi, the Samba package it is going to be used. The goal is to install Samba and use it as a file server to store backups and share files from all the other computers on the network. (Barnes, 2017).

Chapter 2 – System Analysis

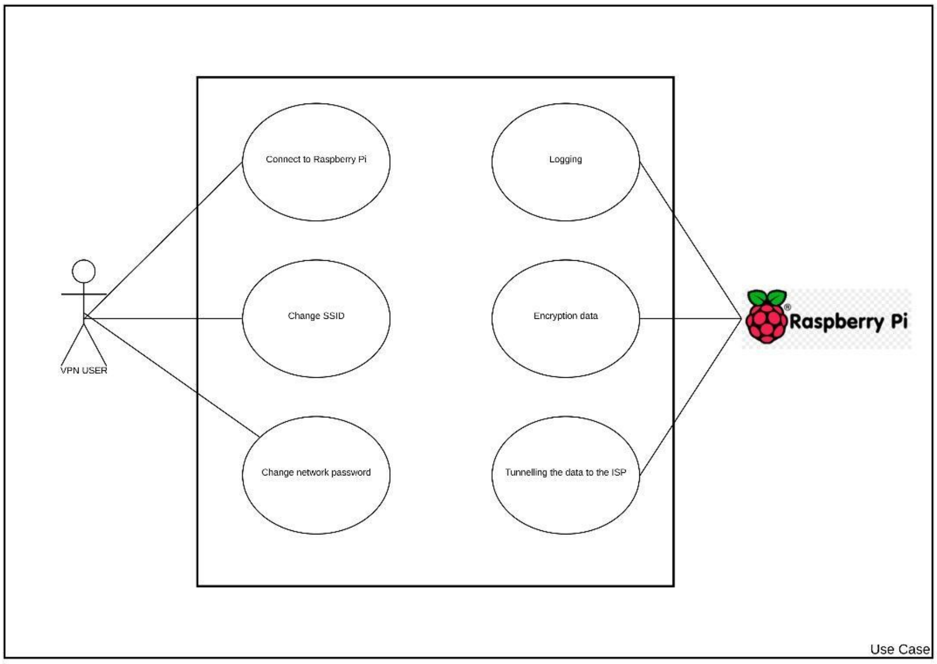
Systems analysis is the process of gathering data, understanding processes, identifying problems and recommending feasible suggestions for improving the system functionalities (Jari, 2020). This involves studying the business processes, collecting operational data, understand the information flow, finding out obstacles and evolving solutions for overcoming the weaknesses of the system in order to achieve the organizational goals.

This project intends to create a private connection between devices located in different networks using the OpenVPN protocol. In order to achieve this, a VPN tunnel has to be created for the network traffic to pass through and cryptographic algorithms have to work to encrypt the data that is transmitted inside the tunnel.

This process can be represented by using some diagrams, such as, Use-Case Diagram, VPN User Diagram, Network Diagram and Sequence Diagram for developed VPN.

Use Case Diagram

Fig. 1 Use-Case Diagram

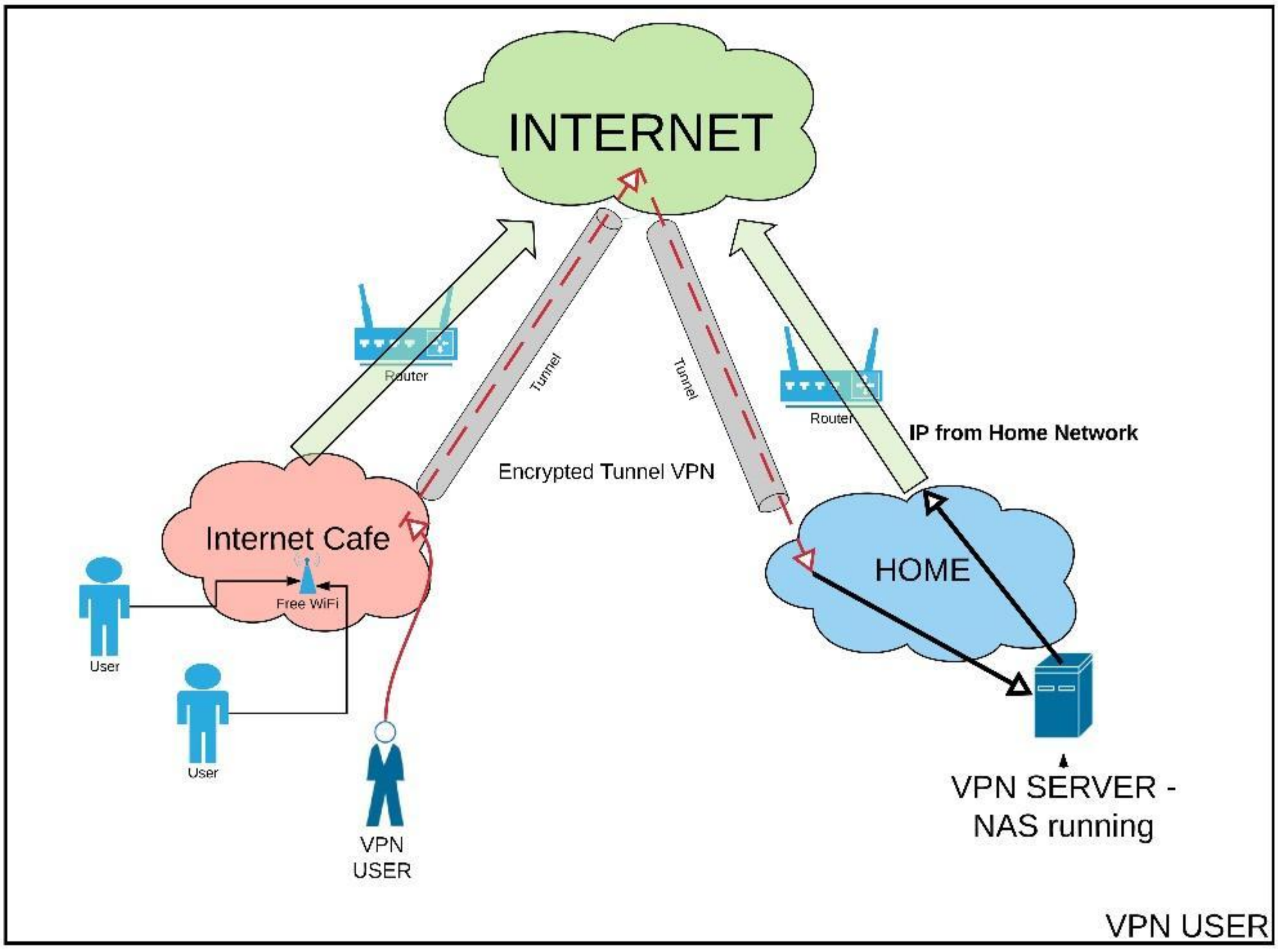


The user's device will connect to the internet and then with Raspberry Pi. The Raspberry pi will encapsulate the traffic by encrypting it through the router. After this process the Raspberry Pi will be able to connect to the internet safely.

Use Case(s) scenarios

Use Case Name:	VPN User Network number 1
Actors:	VPN users
Description:	[VPN users located in a different place from the home network]
Trigger:	[User wants to remotely connect to their home network. The user has set already install a VPN server in the Raspberry Pi.
Preconditions:	Accessing a public internet WIFI there is no guarantee against of cyber-attacks. Using a public IP address the users can be tracked. Travelling abroad, might not be allowed to access social media. Difficulties to bypassing firewall and security servers. Traffic will no longer be unanimous. Files stored at home has no change to be accessed online. Public IP visible while browsing
Postconditions:	Install a VPN server in a Raspberry Pi in order to protect the network traffic inside the tunnel. Lower cost for devices; The dynamic DNS will provide an anonymously static IP address to the user; Data encrypted and access to files that are stored in their home network using NAS. Raspberry Pi hardware will reliably deliver satisfaction Internet speeds; (30mbps cap) Bypass Firewall. VPN tunnel provides anonymity and security while using the Internet. Increase online privacy; More security online allowing users browsing freedom, burying restrictions and online censorships.
Normal Flow:	User without a Raspberry PI VPN connections will be putting their data or personal information in risk while connecting to the public Wi-fi. Without any protections, data and personal information can be in risk of Man in the middle attacks or other cyber-attacks. Some transactions, example Bank transactions, cannot be done while using the public Wi-fi and do not want take risk.

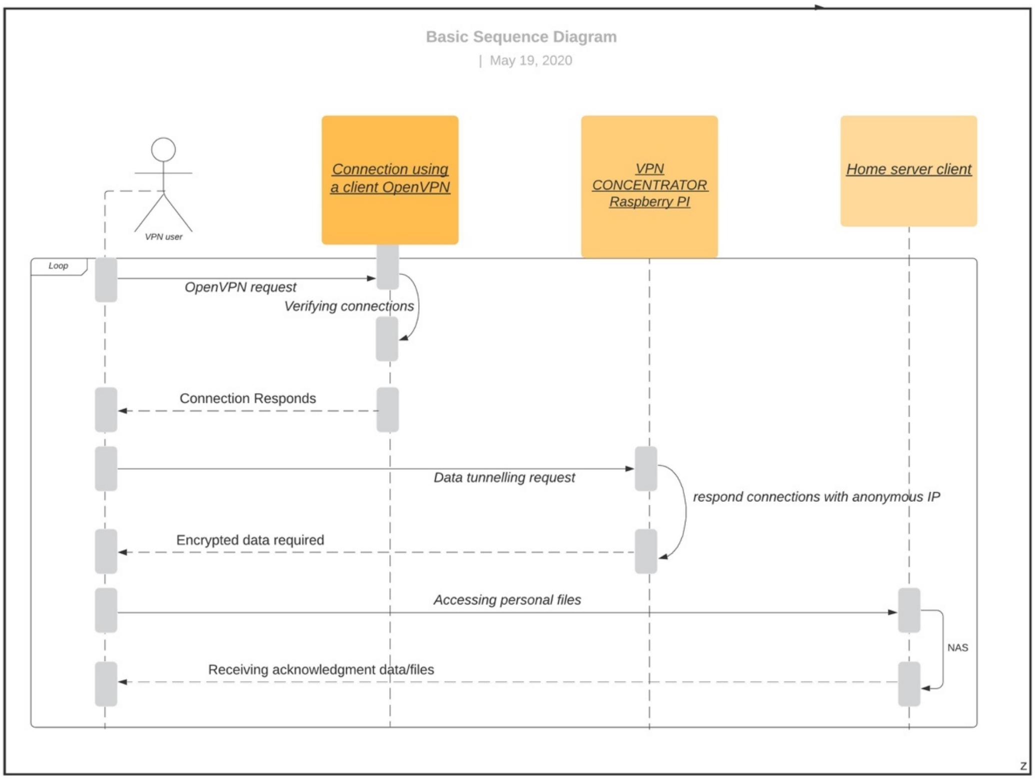
Fig. 2 VPN user



- VPN user will be accessing a public internet Café Wi-fi.
- His connection will be made using an OpenVPN application installed on his phone.
- The request will be encrypted over the internet, encapsulating the package data.
- The home server will be decrypted data packet and check it authentications while it's ok the server will send back a response with a private Network. Although that connections have been made, users will be able accessing files and surf safely through the internet anomaly.

Basic Sequence Diagram

Fig. 3 Sequence Diagram for developed VPN



- User request a VPN connection on a mobile Phone software using openVPN applications.
- Using a generated file key, the VPN user will due a connection through a tunnel to Raspberry Pi located at Home.
- The VPN home NAs will decrypted and check the authentication, while is correct the Raspberry Pi VPN will send back an encrypted data.
- The PI VPN will tunnelling back to the user an anonymous IP address allowing access encrypted data.
- User connected with the raspberry pi are able to use the security encrypted internet data.
- They will be able to view and make changes according password changes an others configurations.

Activity Diagram Connection

Based on Figure 4 Activity Diagram;

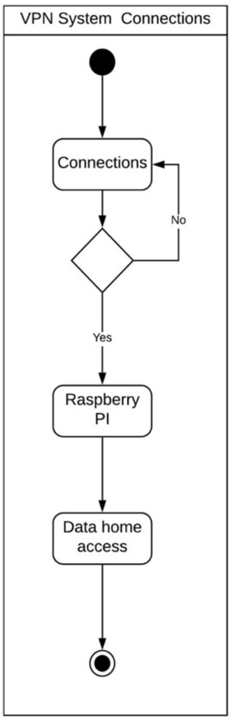
VPN connections activities done by the user accessing the OpenVPN .

The activity diagram is a connection process between user device to the Home Raspberry Pi.

Once the connection between devices are successfully done, the VPN provider will start sending an encrypted data.

While the authentications fails the user will no longer able due a connection between his device and raspberry pi home network.

Fig. 4 Activity Diagram for VPN System Connection



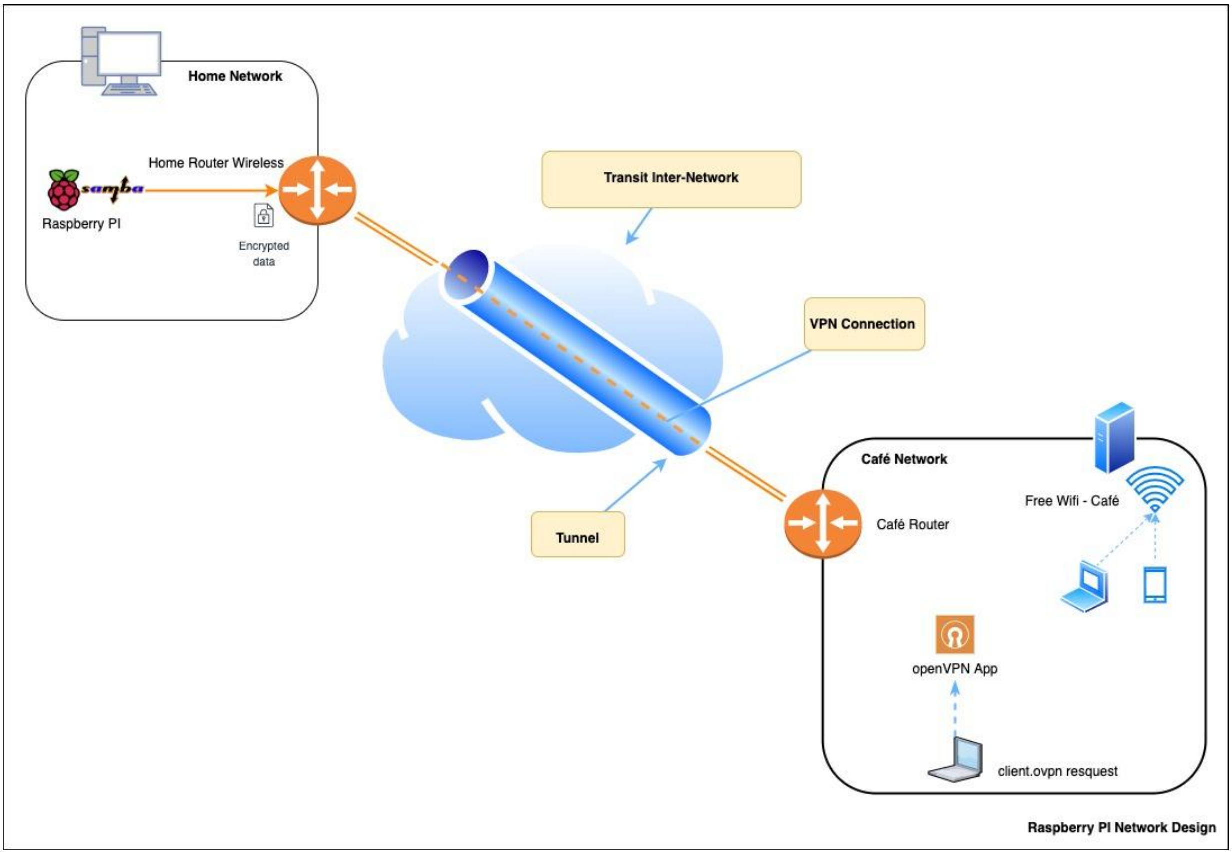
Chapter 3 – System Design

This chapter shows how the system of Virtual Private Network and Network-Attached Storage are going to be going to be implemented in the Raspberry Pi in order to work correctly. For that, some diagrams are going to be shown and some definitions about Port Forwarding, dynamic DNS, PKI and EasyRSA, Firewall (UFW) and Samba.

OpenVPN Network Design

The goal of this project is to install VPN server in the Raspberry Pi and make it works as a Network-Attached Storage (NAS) where files and folders can be accessed remotely and securely from anywhere. In order to achieve that, OpenVPN is going to installed in the Raspberry Pi and then, Samba is going to be installed to enable NAS.

The diagram below illustrates how the system is going to work:



The best way to explain the system would be with an example. Considering a user is connected to a Café Wi-fi network and wants to access his NAS storage located at home. In order to do that, he needs to connect to his home network first.

As previously described network devices can talk to each other using private IP addresses when they are at the same network/subnet. But in our case the user is not in the same network as his NAS.

This is where Open VPN steps in. Using OpenVPN we create a VPN tunnel between the user device located at the Café and the Server (Raspberry Pi) at home. When this VPN connection is established , the user will obtain an IP address assigned to him by OpenVPN server at home (this will be on top of the original IP obtained by Café's network) His device now has an IP in the same range as the NAS server, is part of the same subnet/network and uses the same gateway so it can communicate with all other devices in the same network including the NAS.

Why OpenVPN?

OpenVPN was chosen to be implemented in this project among the others protocols due some characteristics that seemed to be more suitable for the task. It is an open source protocol that it is not difficult to deploy and to be configured. Also, OpenVPN can be deployed in restricted networks. OpenVPN has strong security features that are similar to the ones used in IPsec-based solutions regards to hardware token security and different techniques to authenticate the users (Crist and Keijser, 2015).

The authors state that some of the disadvantages of the use of OpenVPN are the non-existence of a Graphic User interface (GUI) for configuration and management and the necessity of installing the software on the client-side (2015). However, those "disadvantages" are seen as positives, because they brought more challenges to the task since the idea is to configure the VPN through the command line of the Raspberry Pi and not use the GUI.

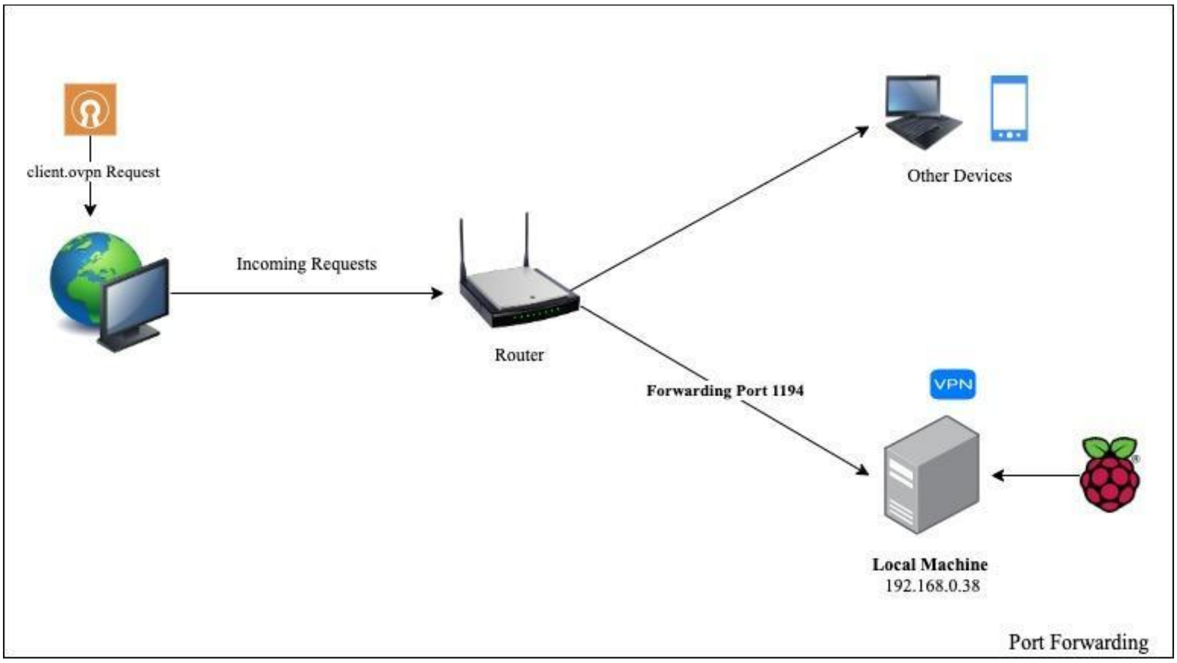
Port Forwarding

Port forwarding is the term used for setting up the router to forward any traffic on a given port to the correct PC in the local network. (VPPPN - Port forwarding, 2020). It is also called "port mapping" and it refers to the network address translator gateway changing the destination address and/or port of the packet to reach a host within a masqueraded, typically private, network. It can be used to allow remote computers to connect to a specific computer within a private network, such as local area network (LAN), in order to make external hosts communicate with services provided by hosts within a LAN. (Ma, 2019)

When the router gets a request from a device on the network, it takes that device and gets the required data from the internet. In order to respond the request, the **ports** are used to guide the router to the computer that generated the request. Therefore, ports are just numbers that are used to "identify" each computer and all network connections need a port. (Hirst, 2018). Port forwarding lets the router match incoming requests with their corresponding port numbers so that the data is forwarded to the right internal computer (Hirst, 2018).

Besides, port forwarding allows unsolicited connections through the NAT firewall on specific ports, making it possible for devices on the internet to initiate connections and access services on a local device (Bischoff, 2019). One of the reasons that VPNs use port forwarding it to make sure the client do not interact directly to the Internet.

Fig. 6 Port Forwarding



This diagram shows an OpenVPN client requesting a home network connection over the internet. The request with a specific addressed port will be sent to the local network. The local networks will get the encrypted request port and process the information. After the keys encrypted match the local machine responds to the request with the network traffic.

Port 1194

Port	Service	Details
1194	OpenVPN	OpenVPN
		OpenVPN(official)

The OpenVPN protocol uses UDP default protocol, however, it is recommended to explicitly mark it in the configuration file to avoid any confusion. This is the local port that OpenVPN listens on. Although, the default port is 1194, any valid and available port number can be assigned for the communication (Crist and Keijser, 2015).

Dynamic DNS

Considering that the public IP address is not static and can change if the router is restarted or if the ISP provider decides, the use of dynamic DNS is essential for the well functionality of the OpenVPN server.

Dynamic DNS (DDNS or DynDNS) is a technique which automatically changes the domain records if there are any changes to the IP address. It is a popular way of handling the network traffic between devices which have IP addresses dynamically assigned by a DHCP server (DNSimple, 2020).

For the implementation of OpenVPN, the NO-IP website is going to be used. It is a free dynamic DNS service provider that allows the users to create a dynamic DNS account to create up to three hostnames without costs. For this project, an account is going to be created to map the hostname to the IP address of the Raspberry Pi.

Samba

The Samba project is one of the most popular ways to turn the Pi into a NAS. It implements the SMB (Server Message Block) and CIFS (Common Internet File System) protocol and works well with most operating systems (Emmet, 2019).

Linux Samba Server is one of the powerful servers that helps you to share files and printers with Windows-based and other operating systems (Dzone, 2017).

Samba can run on many different platforms including Linux, Unix, OpenVMS and operating systems other than Windows and allows the user to interact with a Windows client or server natively. It can basically be described as the Standard Windows interoperability suite of programs for Linux and Unix. (Sohail,2016).

Once set up, you can mount your home file server on all the other computers on your network, and use it as a convenient place to store everything from music files you want to share with your housemates, to backups of important documents and save-game files you'd like to share between computers. . (Barnes, 2017)

Samba is a suite of programs that allows Linux, UNIX, and other systems to interoperate with Microsoft Windows file and printer sharing protocols. Windows, DOS, OS/2, Mac OS/X, and other client systems can access Samba servers to share files and printers in the same ways they would from Windows file and print servers. (Negus and Bresnaham, 2015).

PKI and EasyRSA

The PKI extends for Private key Infrastructure and it was introduced in 1976 to support the asymmetric cryptography. It uses the X.509 standard certificate that it defines the certificate format, it hides the identity of the holder key, etc (smallstep.com, 2018).

In Chapter 1, it was explained that the asymmetric cryptographic uses two different keys: a public key and a private key. The public key is used by the sender to encrypt the message, while the private key is used by the receiver to decrypt the messages. However, before the implantation of certificates, the receiver could not be sure about the security of the key that it received from the sender. In order to solve it, Loren Kohnfelder introduced the idea of certificates to validate the public keys to be transmitted securely over a network (smallstep.com, 2018).

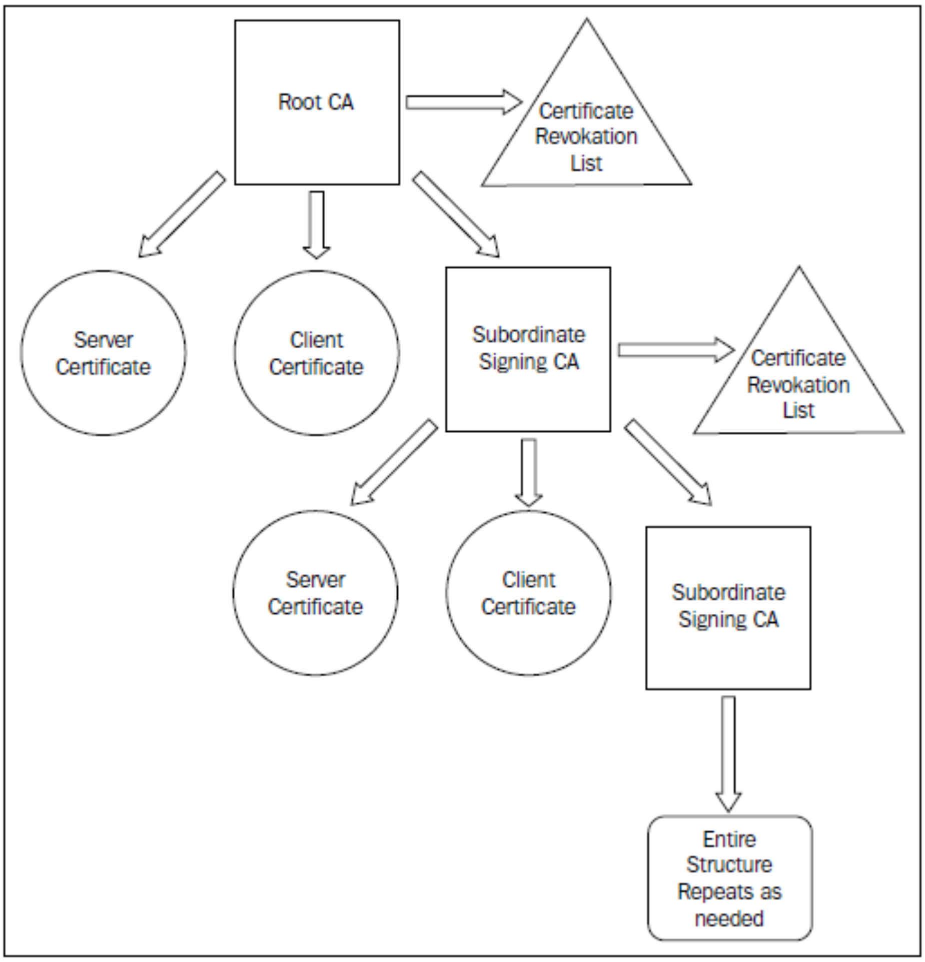
The certificate is a data structure that contains the public key value and the information about the holder of the correspondent private key. Each public key certificate is generated to an individual and each of them have a digital signature of the issuing Certificate Authority. They have an expired data of one to two years and can be revoked in case of loss or private key compromised (author, year).

The PKI validate and authenticate the public key through the use of digital certificates based on the X.509 standard. It organises the certificates and keypairs in a hierarchical way, putting the Certificate Authority on the top of the hierarchy which is the party that is responsible to verify and sign the certificates (Crist and Keisjer, 2015).

The same concept of PKI is used for web browsers, but many users do not know how it works, because the Certificate Authority is already preapproved by the web browser creator or vendor.

A Certificate Authority has to be approved as trust by default and some of its vendors are Go Daddy, Comodo, VeriSign, etc (Crist and Keisjer, 2015).

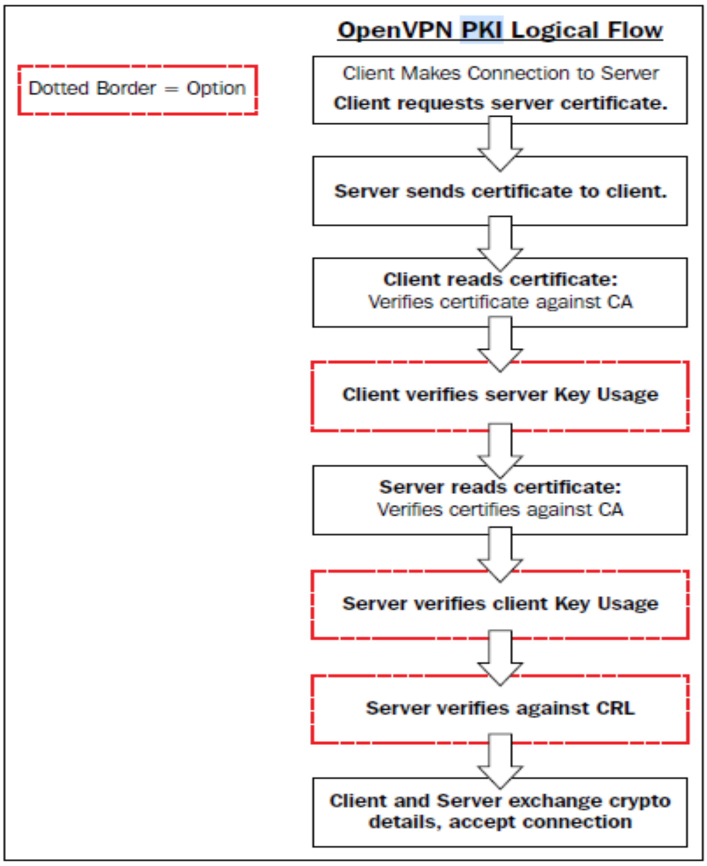
The diagram below shows how the hierarchy of the PKI is organised. It is possible to verify that the CA is the root of the tree. By logic, if the root level is considered to be trusted, all the sub-CA levels are trusted as well.



Crist and Keijser, 2005, p. 64

On this context, the *EasyRSA* is a script which contains configurations files to manage Public Key Infrastructure (PKI) and used to build the Certificates Authorities to issue trusted certificates that encrypt the traffic between the server and the clients (author, year). It creates the PKI with a CA, server certificates, client certificates, etc. which are stored in the EasyRSA directory (Crist and Keisjer, 2015).

The diagram bellow shows the authentication between the server and the client before stabilising a reliable connection. The usage of certificates is the fundamental part of this initial communication.


OpenVPN PKI Logical Flow
Dotted Border = Option
Client Makes Connection to Server
Client requests server certificate.
Server sends certificate to client.
Client reads certificate:
Verifies certificate against CA
Client verifies server Key Usage
Server reads certificate:
Verifies certifies against CA
Server verifies client Key Usage
Server verifies against CRL
Client and Server exchange crypto details, accept connection

Crist and Keijser, 2005, p. 66

One of the advantages of using PKI for OpenVPN it the protection of the VPN, especially when there are many users to connect to the VPN that eventually can have their pre-shared keys stolen or needed to be revoked (Crist and Keijser, 2005).

Firewall (UFW)

A firewall is a set of components or system that is placed between two networks that filters the network traffic based on specific rules. The rules define that only authorized traffic is allowed to pass through it and all the traffic from outside to inside or vice-versa has to go through it (Anicas, 2015).

The firewalls act as a body guard at the entry points of the computers (ports) where the device communicates with others in the network. They inspect the packets that arrive at the ports to ensure they are allowed to go inside or outside the network according to the rules that were set. (Salah-ddine, K, 2017).

In order to implement the VPN in the Raspberry Pi it is necessary to configure the firewall in the device to allow OpenVPN and SSH traffic go through the port 1194 which is the default for OpenVPN. The firewall configuration also includes commands to masquerade any traffic coming from OpenVPN. For that, the Uncomplicated Firewall (UFW) is going to be used. UFW is a Linux package that allow the configuration of firewalls based on the default firewall tool available in Ubuntu (Raspberry Pi Foundation, 2019).

Chapter 4 – Implementation of the System

This chapter shows the implementation of the system. It describes the materials that are necessary to accomplish each task and the steps to install a VPN Server in a Raspberry Pi and to allow the remote access to the devices in the home network. At the end, there is a description of the challenges and problems along the process and how they were solved.

Hardware Requirement

To implement the VPN server oh the Raspberry Pi, many devices were needed, which included router, switches, servers, personal phones and access points. The table below shows the descriptions of all the products:

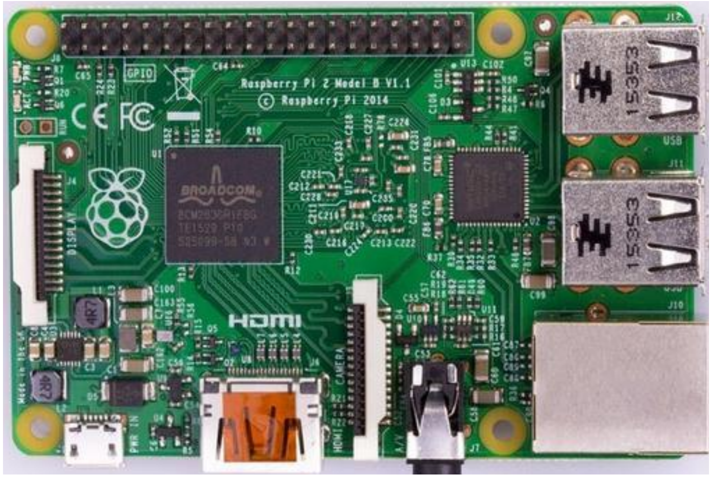
TABLE 1

Hardware Require

Device Name	Model	Quantity	Remark
Router	Cisco Router 819HG – 4G-IOX	1	
Laptop	Windows OS Virtual Box Software	2	Micro SD card HADMI port
Raspberry PI	Raspberry Pi 3 Model B	2	
Micro SD card	Kingston	2	
Monitor		1	HDMI port
Cables	Copper Straight Through, Co-axial	2	
Cables	HDMI cable	1	
Keyboard		1	
Mouse		1	
Phones	Android / IOS	4	

Raspberry Pi technical information:

The Raspberry Pi 3 Model B was the device used to develop the Open VPN to Connect to Home. This generation of Raspberry Pi were released in February 2016 and it allows the Wireless connection. The Raspberry Pi 3 Model B and the digital layout is as shown in Figure 1.



Raspberry Pi 3 Model B have

TABLE 2

RASPBERRY PI CONFIGURATIONS

Processor	A 900MHz quad-core ARM Cortex-A7 CPU
Memory	1GB RAM
Connectivity	2.4GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.1, BLE 4 x USB 2.0 ports
Access	40-pin GPIO 100 Base Ethernet 4 USB ports
Video & Sound	1 x full HDMI Port 1 X DSI display port 1 X CSI camera port Combined 3.5mm audio jack and composite video
SD card support	Micro SD card slot

Installing the OpenVPN server in the Raspberry Pi

Download the installation manager NOOBS (New Out of the Box Software) from the Raspberry Pi website (www.raspberrypi.org/downloads). Drag and drop the files inside the zip folder to the micro Sd card. As simple as that, the micro SD card is ready to be used on the Raspberry Pi. Another alternative is to copy the Raspbian disk image into the micro SD card.

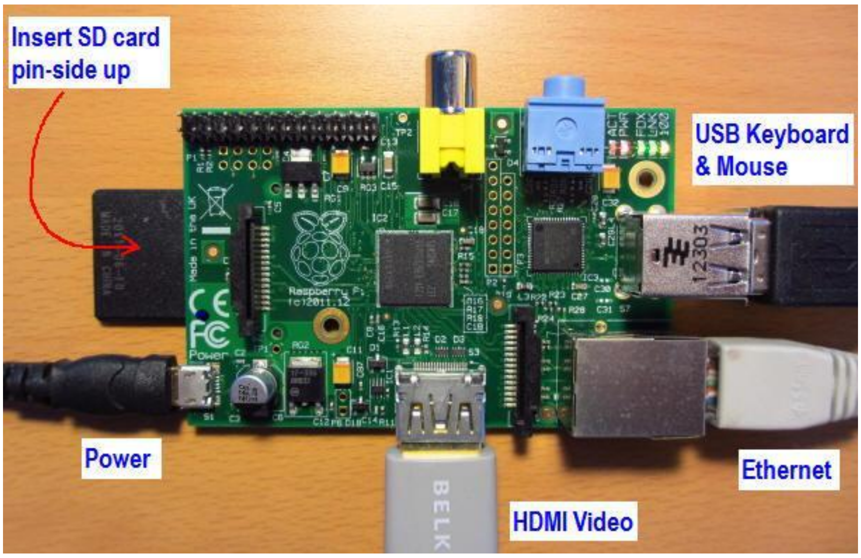
Important note: It is recommended to format the micro SD card to remove old files and folders.

Setting up the raspberry Pi and initial configurations

The first step is to insert the **micro SD** card containing the **Raspbian** operating system installed at the back of the Raspberry Pi. The **keyboard**, **mouse**, **Ethernet cable** should be plugged in the proper ports. The **HDMI cable** should be connected to the **monitor** and to the Raspberry Pi on the correct port. After the cables are plugged in the right places, the Raspberry Pi can be powered on.

A red light should start flash at the beginning and a green light flash when the operating system starts to boot up.

Important note: If the user chooses the Raspberry Pi version 3 is not necessary to use the Ethernet cable, because the device can connect to the network wireless.



Once the operating system is completely booted up, the user can do the initial configurations such as rename the Raspberry Pi, set up a password and enable Secure Shell (SSH). Enabling it allows the user to connect to the Raspberry remotely by using Putty on Windows computers and using the shell for Linux and Mac users.

Initialize the installation of the VPN

Network configuration

The very first task regards the installation of the VPN is to assign a static IP address to the Raspberry Pi. In order to do it, the user should open the command shell in the device and type **ifconfig** and take notes of the following: **[IP Address]** and **[subnet masks].**

After it, type the command: "**netstat -nr**" to obtain information about the router: [**Default Gateway]** and **[Destination].**

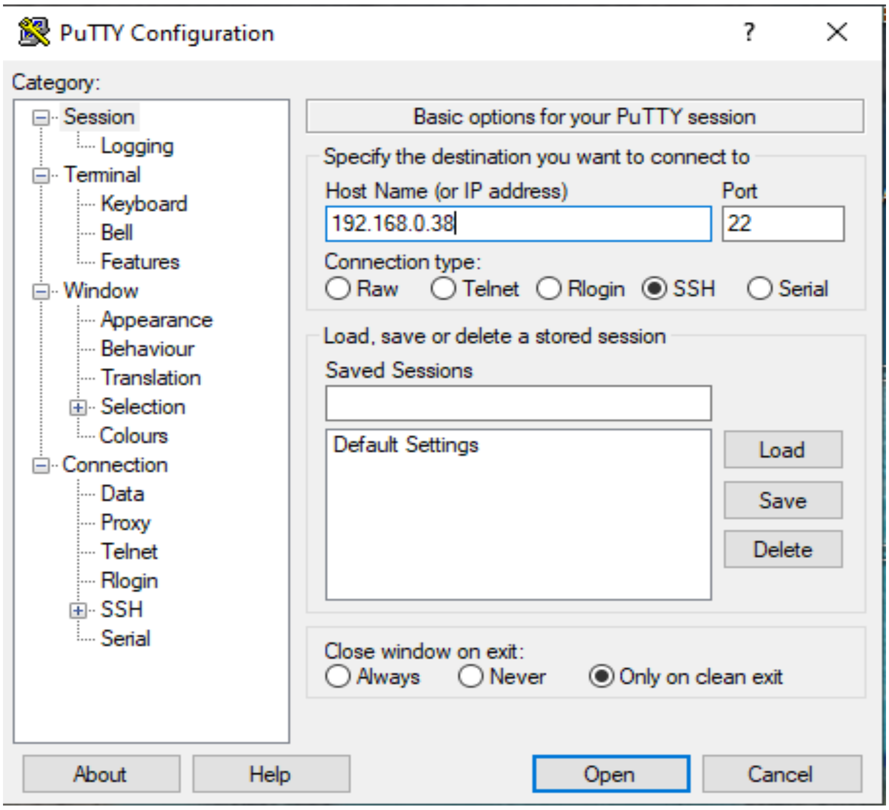
After taking note of those information, the user should edit the ***/etc/dhcpcd.conf*** file by typing the command: **sudo nano /etc/dhcpcd.conf** and add the following lines at the top of the file:

```
Interface wan0(eth0)
Static ip_address= [ip address]/[subnet]
Static routers= [ip address]
Static domain_name_servers= [IP address of the DNS
```

Important Note: The DNS server can be the router address or Google DNS: 8.8.8.8

The user should save the new configuration by pressing [Ctrl + X] – YES and then [Ctrl + O] – exit to go back to the terminal. Reboot the raspberry Pi by typing: **sudo reboot.**

After executing the above configuration, it is possible to access the raspberry pi remotely by using Putty (for users of Windows), so the user does not need the monitor, keyboard and mouse anymore, because the device can be accessed remotely by another machine.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.0.38

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

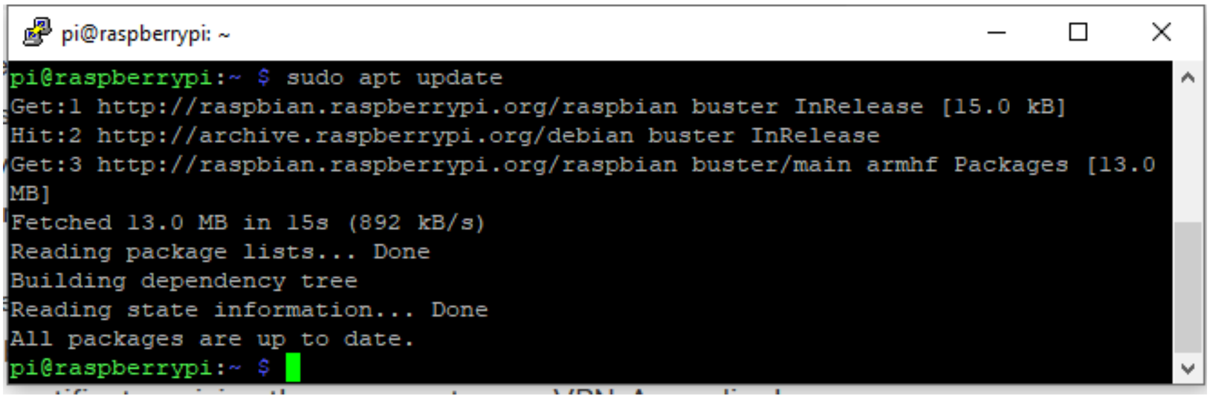
Close window on exit: ○ Always ○ Never ● Only on clean exit

Graphic user interface of the Putty Application

Installing OpenVPN and EasyRSA

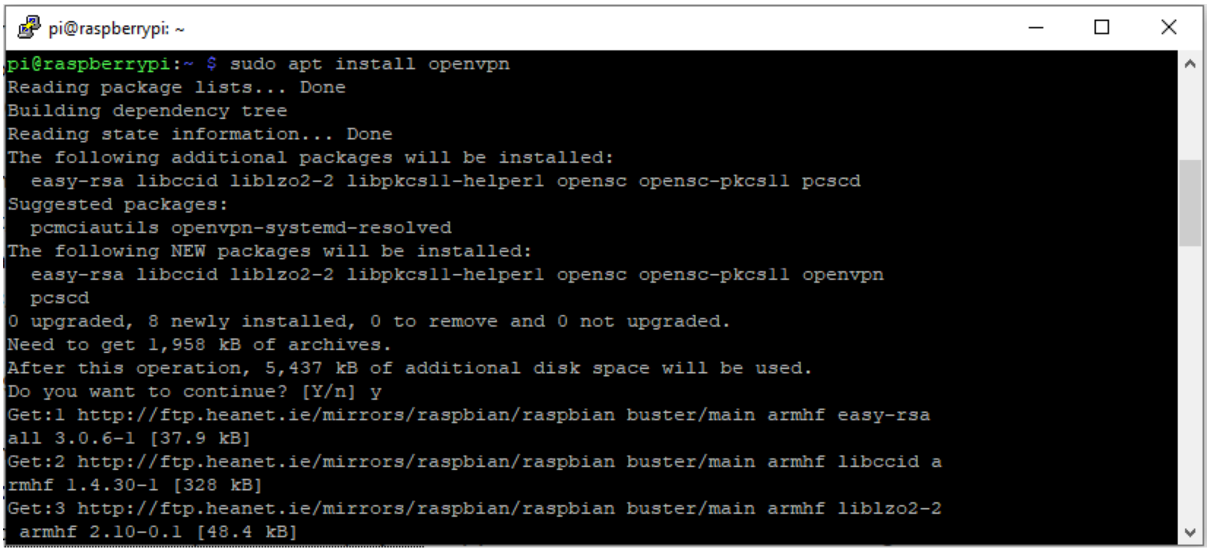
In chapter 3, PKI and EasyRSA were discussed, but just to mention it briefly, the EasyRSA is a script which contains configurations files to manage Public Key Infrastructure (PKI) which is a collection of files that are needed to create Certificate Authorities, keypairs, requests and certificates (EasyRSA, 2020). Meanwhile, Certificate Authorities (CA) are used by OpenVPN to issue trusted certificates that are used to encrypt the traffic between the server and the clients.

Type the command **sudo apt install update**



```
pi@raspberrypi:~ $ sudo apt update
Get:1 http://raspbian.raspberrypi.org/raspbian buster InRelease [15.0 kB]
Hit:2 http://archive.raspberrypi.org/debian buster InRelease
Get:3 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13.0
MB]
Fetched 13.0 MB in 15s (892 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
pi@raspberrypi:~ $
```

Type the command **sudo apt install OpenVPN** to install the OpenVPN server in the Raspberry Pi. After it, the configurations and certificates need to be generated.



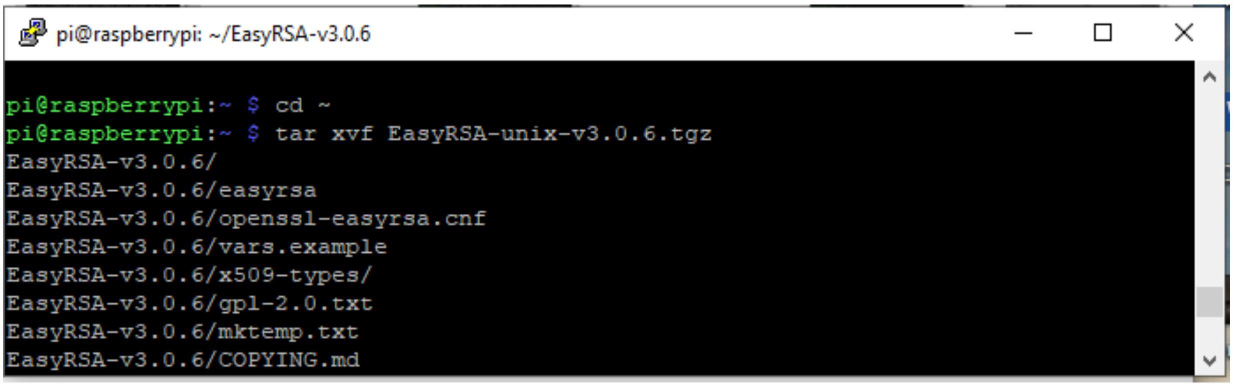
```
pi@raspberrypi:~ $ sudo apt install openvpn
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  easy-rsa libccid liblzo2-2 libpkcs11-helper1 opensc opensc-pkcs11 pcscd
Suggested packages:
  pcmciautils openvpn-systemd-resolved
The following NEW packages will be installed:
  easy-rsa libccid liblzo2-2 libpkcs11-helper1 opensc opensc-pkcs11 openvpn
  pcscd
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,958 kB of archives.
After this operation, 5,437 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.heanet.ie/mirrors/raspbian/raspbian buster/main armhf easy-rsa
all 3.0.6-1 [37.9 kB]
Get:2 http://ftp.heanet.ie/mirrors/raspbian/raspbian buster/main armhf libccid a
rmhf 1.4.30-1 [328 kB]
Get:3 http://ftp.heanet.ie/mirrors/raspbian/raspbian buster/main armhf liblzo2-2
 armhf 2.10-0.1 [48.4 kB]
```

In order to start using the EasyRSA utility it is necessary to download the script from GitHub. For that, type the command to download the EasyRSA script and **wget -P ~/ https://github.com/OpenVPN/easyrsa/releases/download/v3.0.6/EasyRSA-unix-v3.0.6.tgz**

After the download is completed, change the directory to ~ extract the tarball by typing:

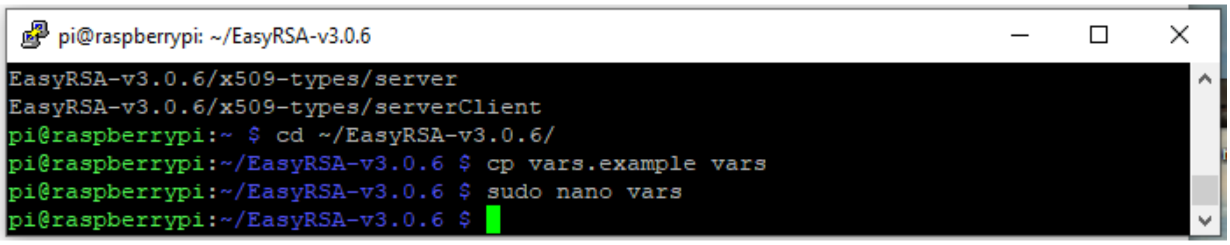
cd ~

tar xvf EasyRSA-unix-v3.0.6.tgz



```
pi@raspberrypi:~ $ cd ~
pi@raspberrypi:~ $ tar xvf EasyRSA-unix-v3.0.6.tgz
EasyRSA-v3.0.6/
EasyRSA-v3.0.6/easyrsa
EasyRSA-v3.0.6/openssl-easyrsa.cnf
EasyRSA-v3.0.6/vars.example
EasyRSA-v3.0.6/x509-types/
EasyRSA-v3.0.6/gpl-2.0.txt
EasyRSA-v3.0.6/mktemp.txt
EasyRSA-v3.0.6/COPYING.md
```

Go to the ~/EasyRSA directory by typing **cd ~/EasyRSA-v3.0.6/** and make a copy of the file **vars.example** naming it **vars.** Type: **cp vars.example vars.** Then, type **sudo nano vars** to edit the file.

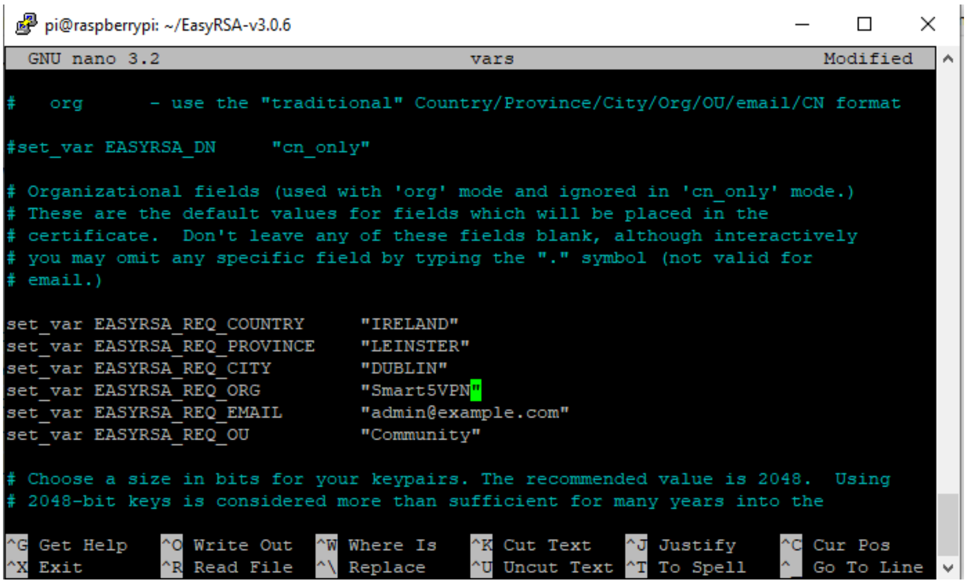


```
EasyRSA-v3.0.6/x509-types/server
EasyRSA-v3.0.6/x509-types/serverClient
pi@raspberrypi:~ $ cd ~/EasyRSA-v3.0.6/
pi@raspberrypi:~/EasyRSA-v3.0.6 $ cp vars.example vars
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo nano vars
pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

Inside the vars file, find the following section and uncomment the lines by erasing the # sign and replace the fields with your own information. After it, save (CTRL + X) and close the file.

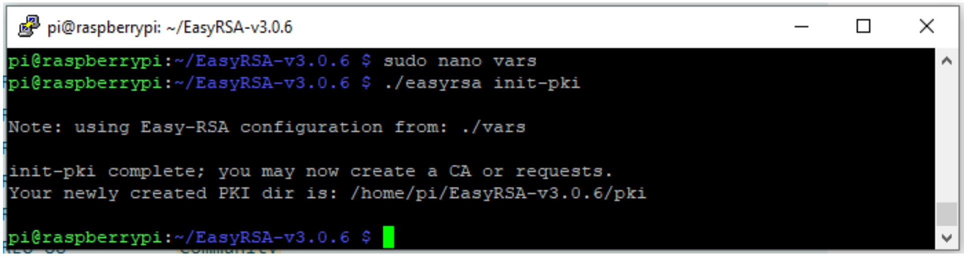
~/EasyRSA-v3.0.6/vars

```
. . .

#set_var EASYRSA_REQ_COUNTRY    "US"
#set_var EASYRSA_REQ_PROVINCE   "California"
#set_var EASYRSA_REQ_CITY       "San Francisco"
#set_var EASYRSA_REQ_ORG        "Copyleft Certificate Co"
#set_var EASYRSA_REQ_EMAIL      "me@example.net"
#set_var EASYRSA_REQ_OU         "My Organizational Unit"

. . .
```



```
  GNU nano 3.2                          vars                           Modified

#   org      - use the "traditional" Country/Province/City/Org/OU/email/CN format

#set_var EASYRSA_DN     "cn_only"

# Organizational fields (used with 'org' mode and ignored in 'cn_only' mode.)
# These are the default values for fields which will be placed in the
# certificate.  Don't leave any of these fields blank, although interactively
# you may omit any specific field by typing the "." symbol (not valid for
# email.)

set_var EASYRSA_REQ_COUNTRY     "IRELAND"
set_var EASYRSA_REQ_PROVINCE    "LEINSTER"
set_var EASYRSA_REQ_CITY        "DUBLIN"
set_var EASYRSA_REQ_ORG         "Smart5VPN"
set_var EASYRSA_REQ_EMAIL       "admin@example.com"
set_var EASYRSA_REQ_OU          "Community"

# Choose a size in bits for your keypairs. The recommended value is 2048.  Using
# 2048-bit keys is considered more than sufficient for many years into the

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

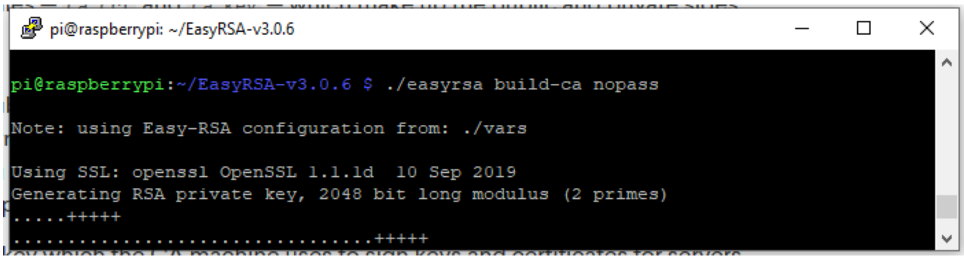
Run the script **EasyRSA** which is inside the EasyRSA directory to manage and build the certificate authority. The script has to be run with the **init pki** option in order to initiate the public key infrastructure on the CA server (Ellingwood and Drake, 2019). Type the command **./easyrsa init-pki**



```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo nano vars
pi@raspberrypi:~/EasyRSA-v3.0.6 $ ./easyrsa init-pki

Note: using Easy-RSA configuration from: ./vars

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/pi/EasyRSA-v3.0.6/pki

pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

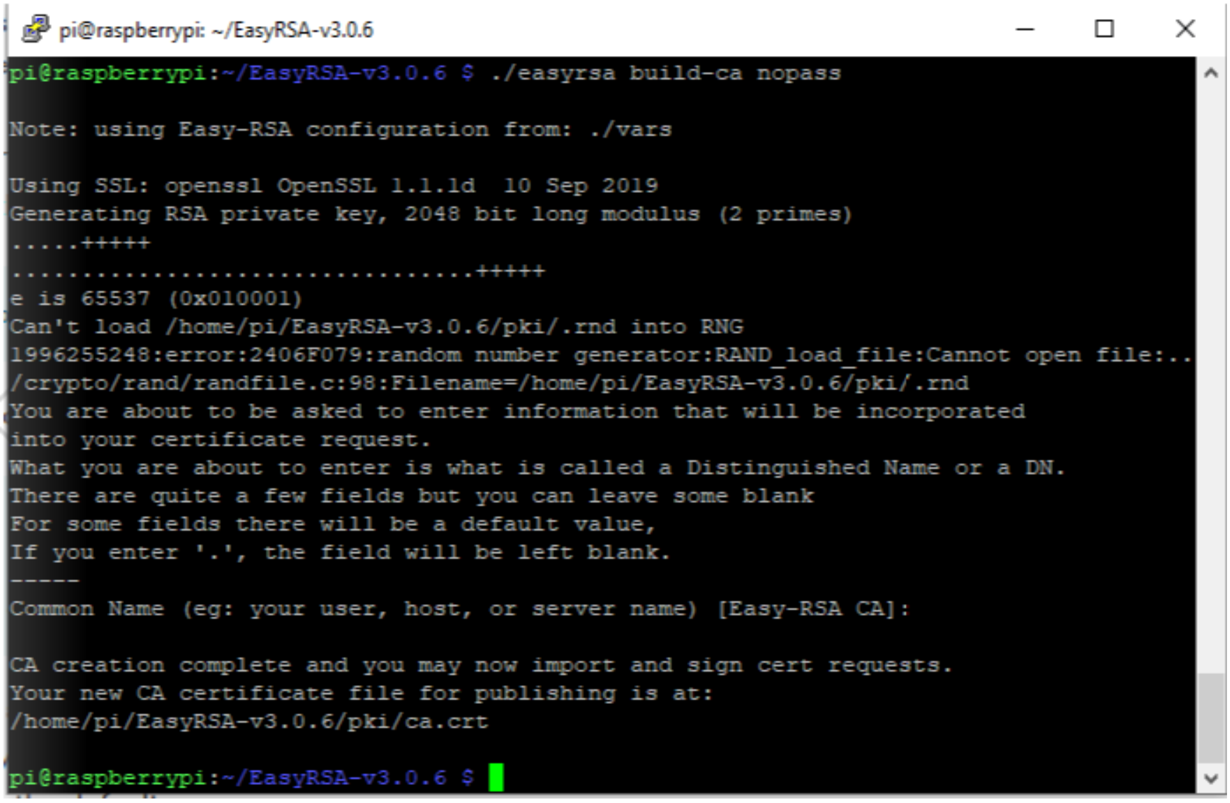
The CA can now start to be build and the files **ca.crt** and **ca.key** are going to be created. They compound the public and private keys of the certificates. The **ca.crt** is the public certificate file which is used by the server and the client to inform each other they are part of the same web of trust. Meanwhile, the **ca.key** is the private key used to sign keys and certificates to servers and clients (Ellingwood and Drake, 2019). For that, the command **./easyrsa build-ca nopass** has to be issued.

Important note: the nopass parameter means that no password needs to be created. If you want to insert a password, do not add it at the end.



```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ ./easyrsa build-ca nopass

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.................................+++++
```

The user will be asked to confirm the common name for the CA. Just press ENTER to leave as the default. If the user wants to change the name, it needs to pay attention and change the names on the next steps when generating the other keys.

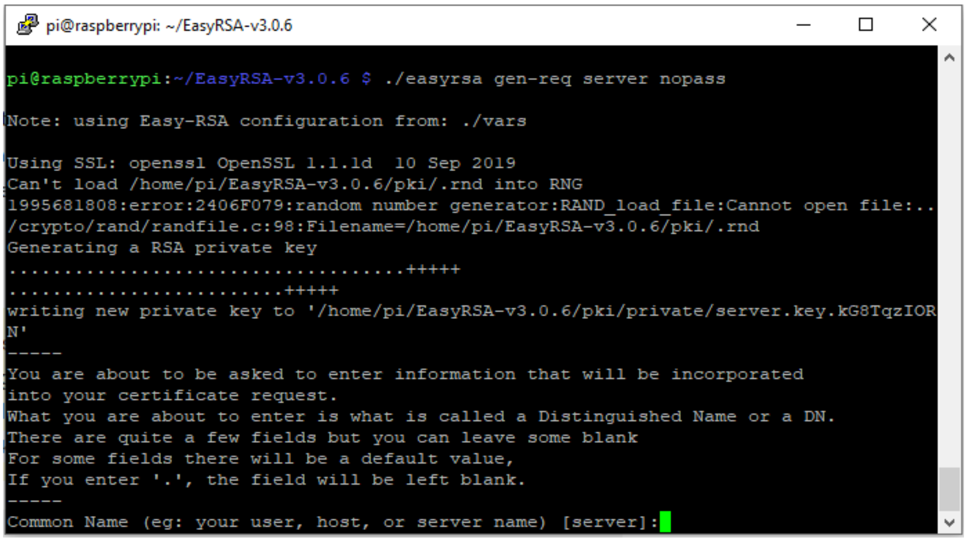


```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ ./easyrsa build-ca nopass

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.................................+++++
e is 65537 (0x010001)
Can't load /home/pi/EasyRSA-v3.0.6/pki/.rnd into RNG
1996255248:error:2406F079:random number generator:RAND_load_file:Cannot open file:..
/crypto/rand/randfile.c:98:Filename=/home/pi/EasyRSA-v3.0.6/pki/.rnd
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/home/pi/EasyRSA-v3.0.6/pki/ca.crt

pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

The **ca.crt** file was generated in the directory: /home/pi/EasyRSA-v3.0.0/pki/ca.crt

Create certificates and key files for the server

Type the command **./easyrsa gen-req server nopass** to generate a private key for the server and a certificate request file named **server.req (**Ellingwood and Drake, 2019). The common name will be named server, but the user can change it if they want. Or just press ENTER to leave as server.



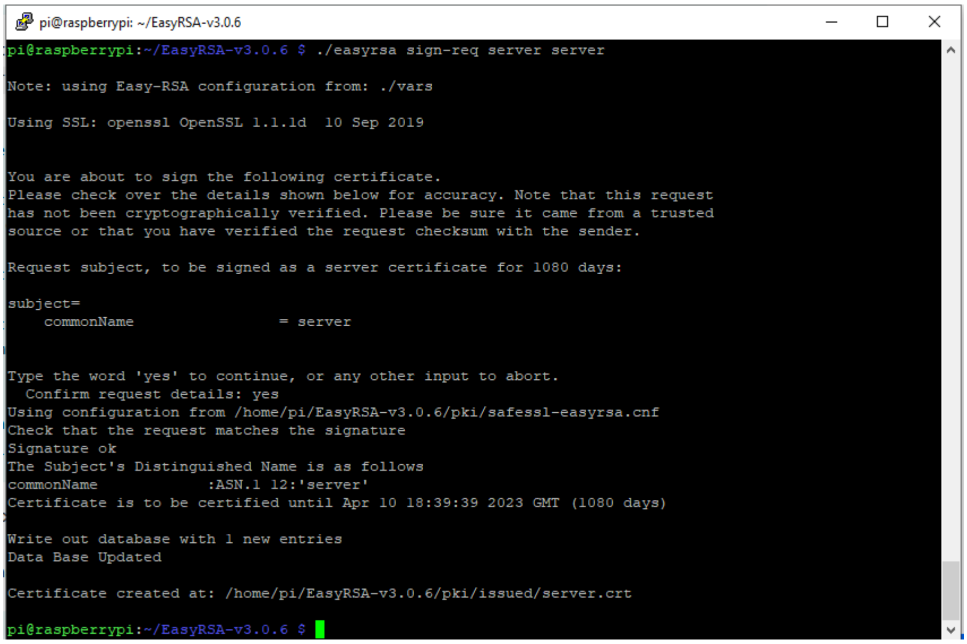
pi@raspberrypi: ~/EasyRSA-v3.0.6

```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ ./easyrsa gen-req server nopass

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019
Can't load /home/pi/EasyRSA-v3.0.6/pki/.rnd into RNG
1995681808:error:2406F079:random number generator:RAND_load_file:Cannot open file:..
/crypto/rand/randfile.c:98:Filename=/home/pi/EasyRSA-v3.0.6/pki/.rnd
Generating a RSA private key
...................................+++++
........................+++++
writing new private key to '/home/pi/EasyRSA-v3.0.6/pki/private/server.key.kG8TqzIOR
N'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [server]:
```

The file needs to be copied to a folder inside the OpenVPN directory (/etc/openvpn/). For that, type **sudo cp ~/EasyRSA-v3.0.6/pki/server.key /etc/openvpn/**

Important note: pay attention to the path of the server.key file because sometimes the key is generated in other location.

Run the **easyrsa** script again using **sign-req** option to sign the request for the server certificate by typing the command **./easyrsa sign-req server server** which includes the request type '[server]' and the common name '[server]'. A message will ask for confirmation, so just type **yes.**



pi@raspberrypi: ~/EasyRSA-v3.0.6

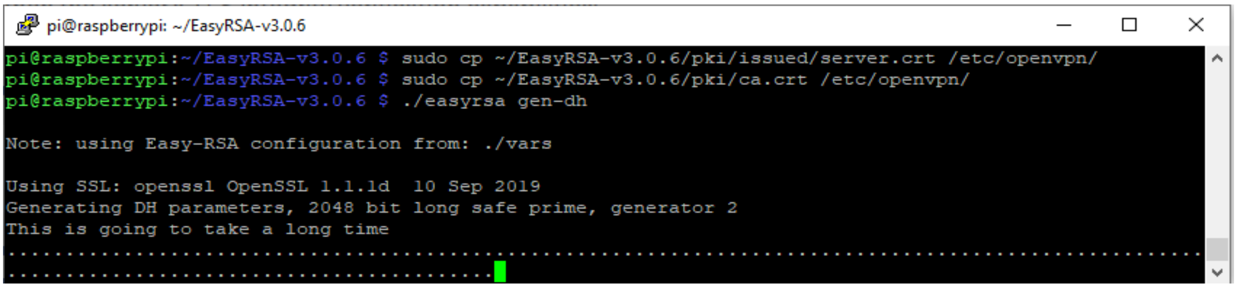
```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ ./easyrsa sign-req server server

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019


You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 1080 days:

subject=
    commonName                = server


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from /home/pi/EasyRSA-v3.0.6/pki/safessl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'server'
Certificate is to be certified until Apr 10 18:39:39 2023 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/pi/EasyRSA-v3.0.6/pki/issued/server.crt

pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

Next the keys, **server.crt** and **ca.crt** should be copied to the **/etc/openvpn/** directory. For that, you must check the location path for both keys and type the copy command.

sudo cp ~EasyRSA-v3.0.6/issued/server.crt /etc/openvpn/

sudo cp ~/EasyRSA-v3.0.6/pki/ca.crt /etc/openvpn/

Then, run the **./easyrsa** script using **gen-dh** to generate a Diffie-Helman key by typing **sudo ./easyrsa gen-dh.** The Diffie-Helman key is required for the VPN session keys which are temporary and generated when the connection between the client and the server happens for the first time. In other words, this file allows the server to establish a secure TLS connection with the client (Crist and Keijser, 2015).



```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo cp ~/EasyRSA-v3.0.6/pki/issued/server.crt /etc/openvpn/
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo cp ~/EasyRSA-v3.0.6/pki/ca.crt /etc/openvpn/
pi@raspberrypi:~/EasyRSA-v3.0.6 $ ./easyrsa gen-dh

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
..........................................................................................
.........................................
```

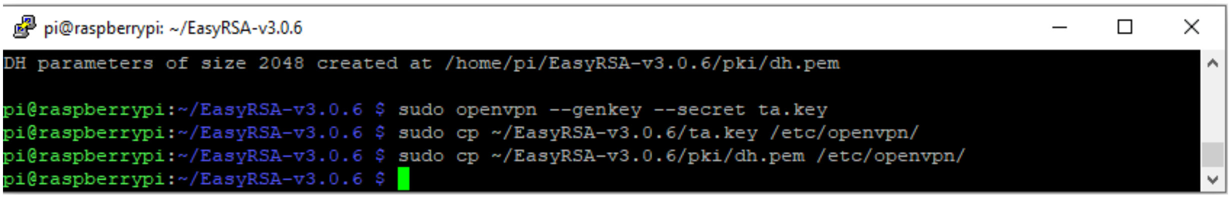
After it, a secret key need to be generated to secure the OpenVPN connection. The reason is that OpenVPN server establish an TLS control channel for each client that tries to establish a connection and by doing so, it avoids **denial-of-service-attacks.** This key adds an extra layer to the TLS channel authentication between the clients and the server and avoids this type of attack (Crist and Keijser, 2015).

sudo openvpn --genkey --secret ta.key

After it, the **dh key** and **ta.key** needs to be copied to the **/etc/openvpn/** directory by typing

sudo cp ~/EasyRSA-v3.0.6/ta.key /etc/openvpn/

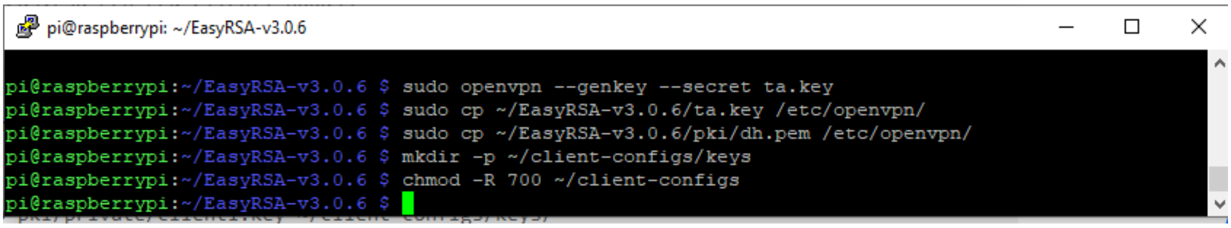
sudo cp ~/EasyRSA-v3.0.6/pki/dh.pem /etc/openvpn/



```
DH parameters of size 2048 created at /home/pi/EasyRSA-v3.0.6/pki/dh.pem

pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo openvpn --genkey --secret ta.key
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo cp ~/EasyRSA-v3.0.6/ta.key /etc/openvpn/
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo cp ~/EasyRSA-v3.0.6/pki/dh.pem /etc/openvpn/
pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

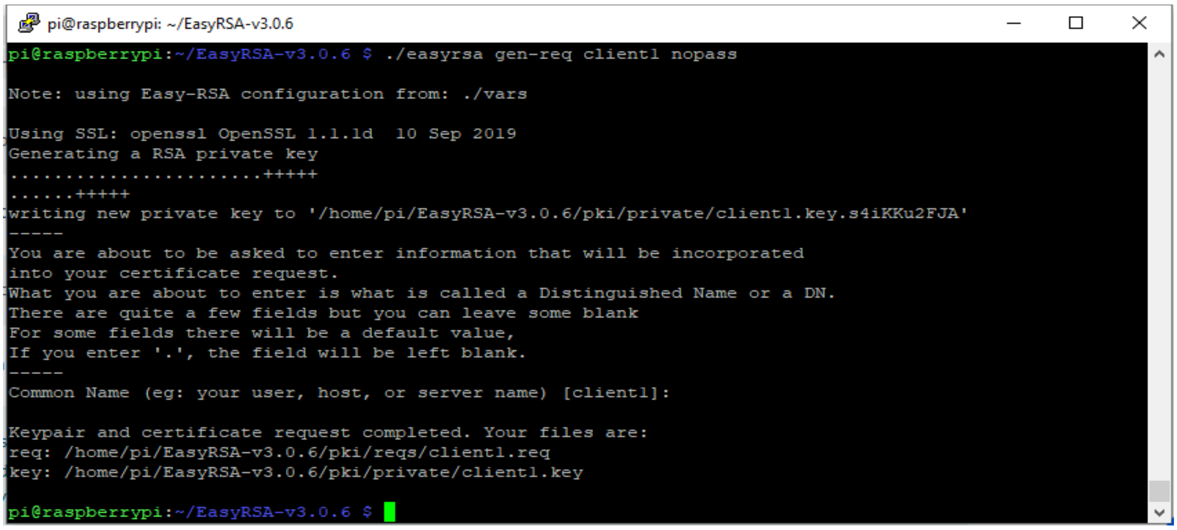
Generate certificates and key files for the client

Create a directory where the client keys and certificates will be stored by typing **mkdir -p ~/client-configs/keys** and change the permissions of the directory for secure measures **sudo chmod -R 700 ~/client-configs**



```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo openvpn --genkey --secret ta.key
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo cp ~/EasyRSA-v3.0.6/ta.key /etc/openvpn/
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo cp ~/EasyRSA-v3.0.6/pki/dh.pem /etc/openvpn/
pi@raspberrypi:~/EasyRSA-v3.0.6 $ mkdir -p ~/client-configs/keys
pi@raspberrypi:~/EasyRSA-v3.0.6 $ chmod -R 700 ~/client-configs
pi@raspberrypi:~/EasyRSA-v3.0.6 $ 
```

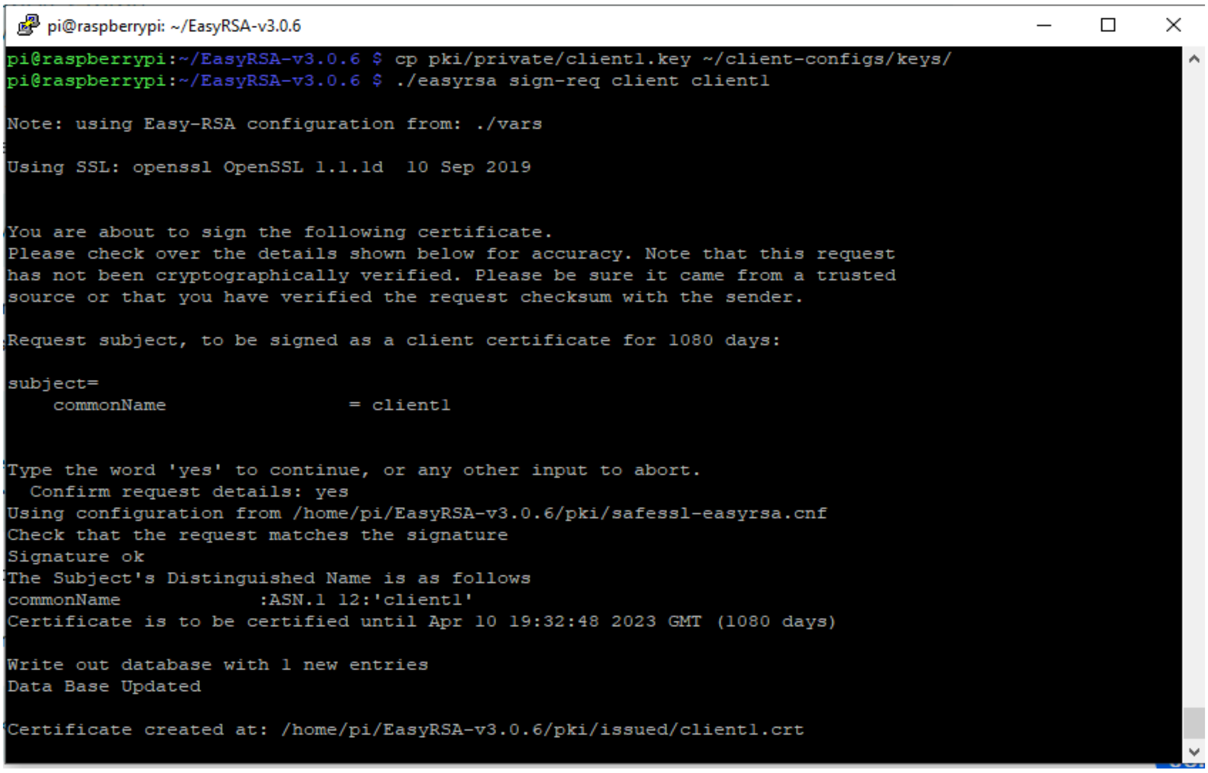
Next, run the **easyrsa** script again using **gen-req** to generate the certificate for the client by typing **./easyrsa gen-req client1 nopass** and press ENTER to agree with the common name.



```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ ./easyrsa gen-req client1 nopass

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019
Generating a RSA private key
.......................+++++
......+++++
writing new private key to '/home/pi/EasyRSA-v3.0.6/pki/private/client1.key.s4iKKu2FJA'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [client1]:

Keypair and certificate request completed. Your files are:
req: /home/pi/EasyRSA-v3.0.6/pki/reqs/client1.req
key: /home/pi/EasyRSA-v3.0.6/pki/private/client1.key

pi@raspberrypi:~/EasyRSA-v3.0.6 $ 
```

Once the keypair and certificate request were generated copy the **client key** to the **~/client-configs/keys/** directory by typing **sudo cp ~/EasyRSA-v3.0.6/pki/private/client1.key ~/client-configs/keys/**

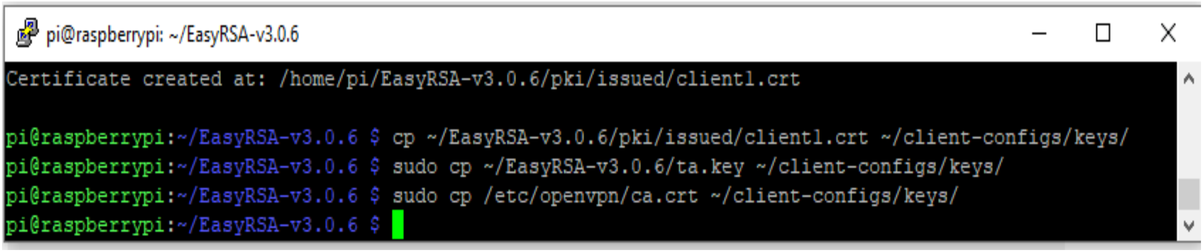
Then sign the request to the client by running the e**asyrsa** script specifying the client request. Type **./easyrsa sign-req client client1.** When the prompt asks for confirmation, type **yes**.



pi@raspberrypi: ~/EasyRSA-v3.0.6

```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ cp pki/private/client1.key ~/client-configs/keys/
pi@raspberrypi:~/EasyRSA-v3.0.6 $ ./easyrsa sign-req client client1

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019


You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a client certificate for 1080 days:

subject=
    commonName                = client1


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from /home/pi/EasyRSA-v3.0.6/pki/safessl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'client1'
Certificate is to be certified until Apr 10 19:32:48 2023 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/pi/EasyRSA-v3.0.6/pki/issued/client1.crt
```

After the creation of the **client1.cert file,** copy the client certificate to the **~/client-configs/keys/ directory.** Along with the client certificate, the files ca.crt and ta.key should be copied to the **~/client-configs/keys/** directory.



pi@raspberrypi: ~/EasyRSA-v3.0.6

```
Certificate created at: /home/pi/EasyRSA-v3.0.6/pki/issued/client1.crt

pi@raspberrypi:~/EasyRSA-v3.0.6 $ cp ~/EasyRSA-v3.0.6/pki/issued/client1.crt ~/client-configs/keys/
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo cp ~/EasyRSA-v3.0.6/ta.key ~/client-configs/keys/
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo cp /etc/openvpn/ca.crt ~/client-configs/keys/
pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

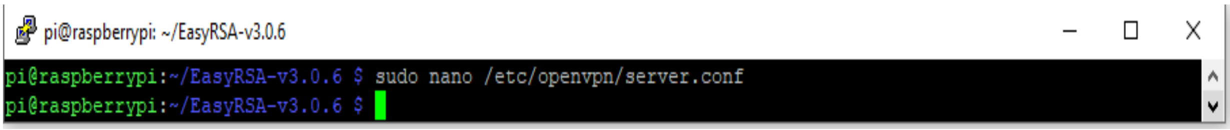
All the clients and server keys and certificates were generated.

Configuration of OpenVPN

The first step is to copy the OpenVPN configurations sample file into the configuration directory by typing the command **sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/**

Extract the files from the folder by typing the command **sudo gzip -d /etc/openvpn/server.conf.gz**

After it, open the server configuration file in a text editor by typing **sudo nano /etc/openvpn/server.conf.**

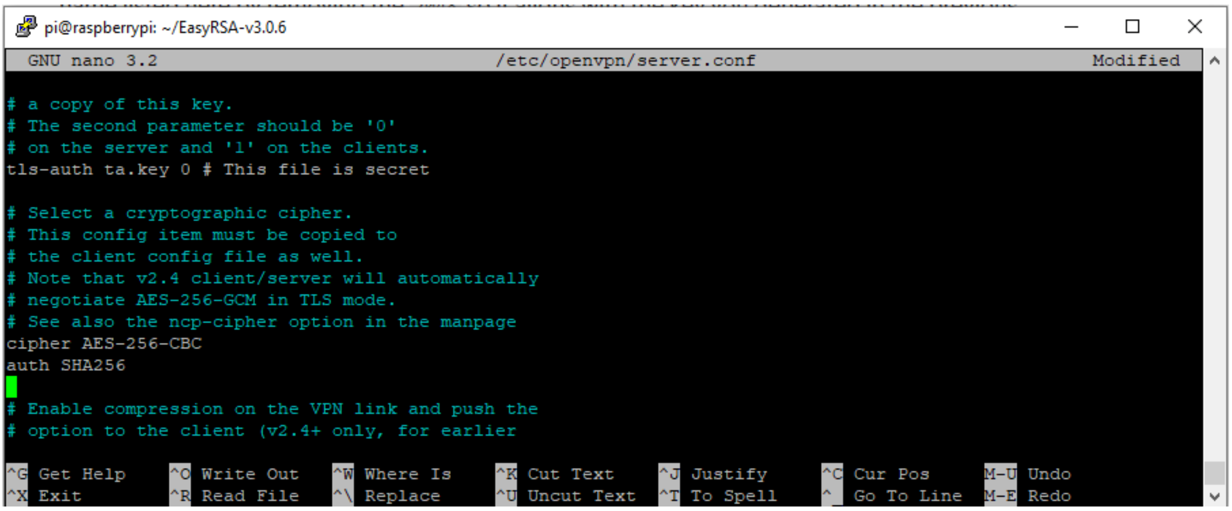


pi@raspberrypi: ~/EasyRSA-v3.0.6

```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo nano /etc/openvpn/server.conf
pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

Inside the file, go to the HMA section and uncomment the line **tls-auth ta.key 0 # This file is secret.** It supposed to be uncommented, but in case it is not, just remove the '#' or '**;**' sign from the beginning. The number **0** in the command represents a direction flag that signalises that different keys are going to be used to encrypt and to decrypt the data. It is set to 0 in one end (server) and it is set to 1 on the other end (client) (Crist and Keijser, 2015).

Next, got to the cypher section and uncomment the line **cipher AES-256-CBC.** The line should be uncommented already. Under the cypher line, add the command: **auth SHA256.** Those two lines are the ones that define the cryptographic cipher.

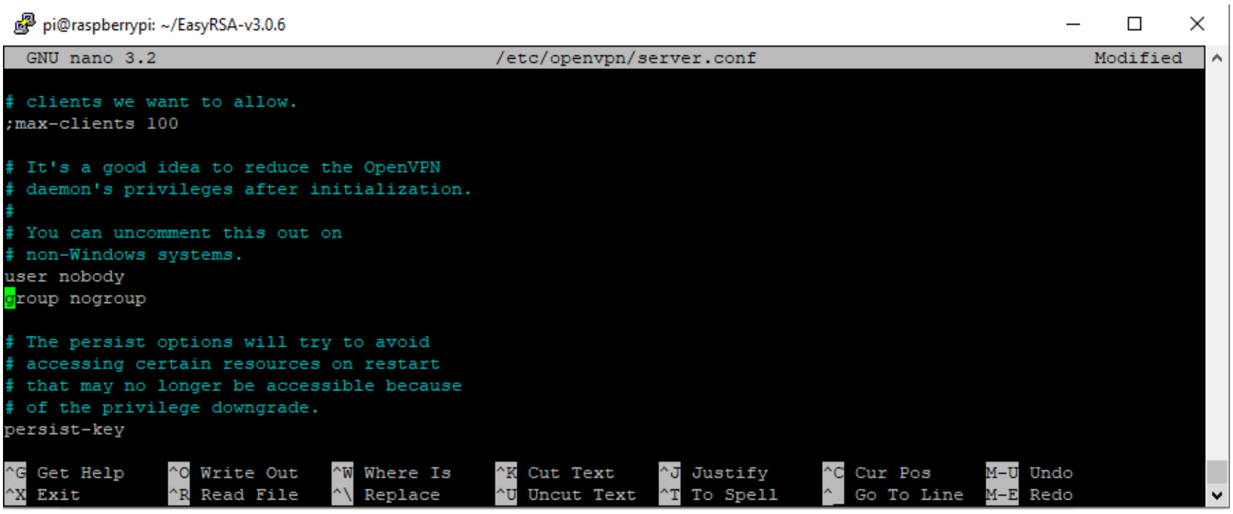


pi@raspberrypi: ~/EasyRSA-v3.0.6

```
  GNU nano 3.2                    /etc/openvpn/server.conf                    Modified

# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
tls-auth ta.key 0 # This file is secret

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
cipher AES-256-CBC
auth SHA256

# Enable compression on the VPN link and push the
# option to the client (v2.4+ only, for earlier

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line M-E Redo
```

Then, go to the **dh** line that defines the Diffie-Hellman parameters and change the name for **dh dh.pem** by removing the 2048 from the name of the file. This change is necessary to adequate the name of the file generated in the previous step to the file in the configuration file **(**Ellingwood and Drake, 2019).

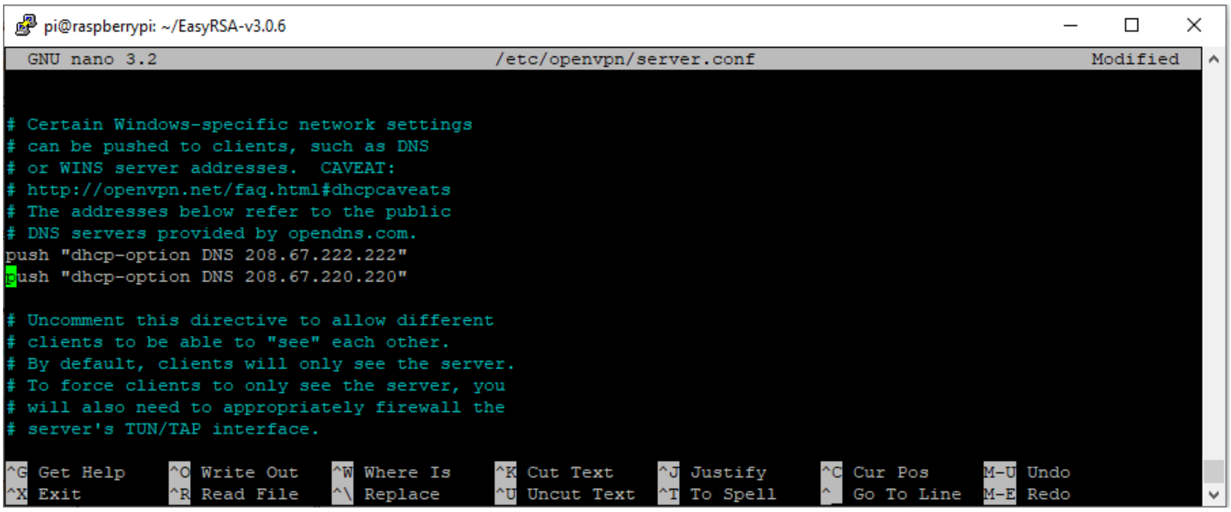
Go to the users and groups settings and uncomment the two lines **user nobody** and **group nogroup.**



```
  GNU nano 3.2                  /etc/openvpn/server.conf                  Modified

# clients we want to allow.
;max-clients 100

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
user nobody
group nogroup

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line M-E Redo
```

In order to have all the traffic going through the VPN the DNS settings need to be push to the client machines. So, find the **redirect-gateway** section and uncomment the line push **"redirect-gateway def1 bypass-dhcp".**

Go to the next section **dhcp-option** and uncomment the following lines **push "dhcp-option DNS 208.67.222.222"** and **push "dhcp-option DNS 208.67.220.220".** Those configurations are important to configure the DNS settings in the client machine to use the VPN tunnel as the default gateway (Ellingwood and Drake, 2019).

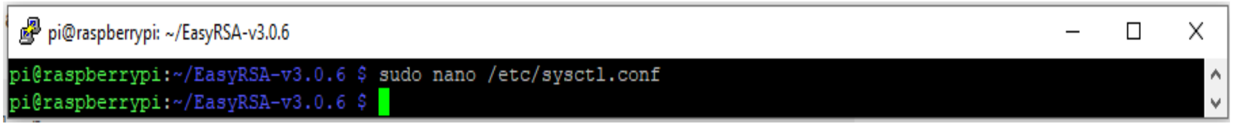


```
pi@raspberrypi: ~/EasyRSA-v3.0.6
  GNU nano 3.2                    /etc/openvpn/server.conf                    Modified

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  M-E Redo
```

After that, save the file (Crtl + X) and exit.

OpenVPN network configuration

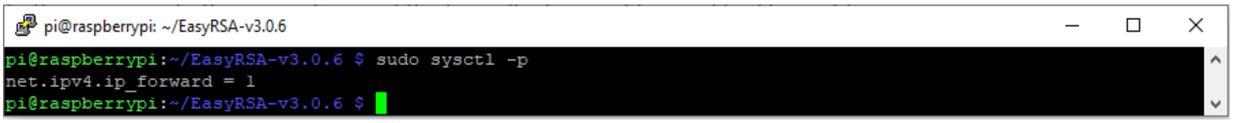
The first step is to open the **systcl.conf** file using a text editor to add some settings to the file. Type **sudo nano /etc/sysctl.conf.**



```
pi@raspberrypi: ~/EasyRSA-v3.0.6
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo nano /etc/sysctl.conf
pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

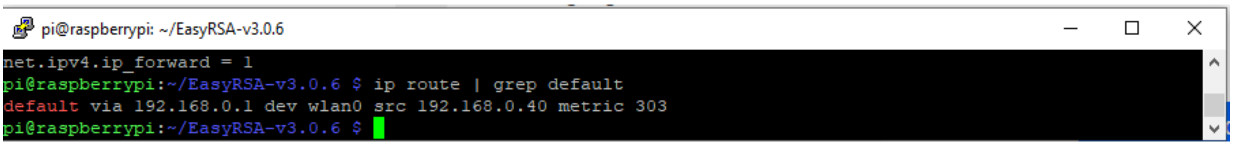
Inside the file, uncomment the line **net.ipv4.ip_foward=1** to enable port forwarding. Port forwarding, as it was mentioned on previous chapters, allows that enables the port 1194 to receive and forward OpenVPN network traffic to the Raspberry Pi.

Type the command **sysctl -p** to confirm the alteration and to apply the changes without having to reboot the operating system. The output should be **net.ipv4.ip_foward=1**



```
pi@raspberrypi: ~/EasyRSA-v3.0.6
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo sysctl -p
net.ipv4.ip_forward = 1
pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

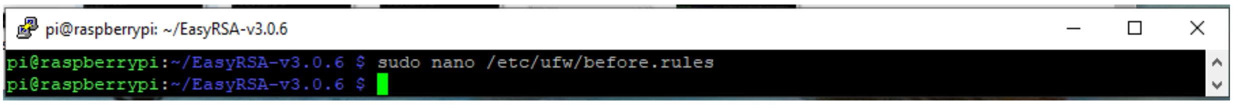
The next step is to configure the firewall. Before, it is necessary to find out the public network interface in the raspberry pi by typing the command **ip route | grep default.**



```
net.ipv4.ip_forward = 1
pi@raspberrypi:~/EasyRSA-v3.0.6 $ ip route | grep default
default via 192.168.0.1 dev wlan0 src 192.168.0.40 metric 303
pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

The following commands are needed to configure the firewall in order to allow and masquerade traffic from OpenVPN. The uncomplicated firewall (UFW) is the default firewall in Ubuntu, but that can be easily used in the Raspberry Pi.

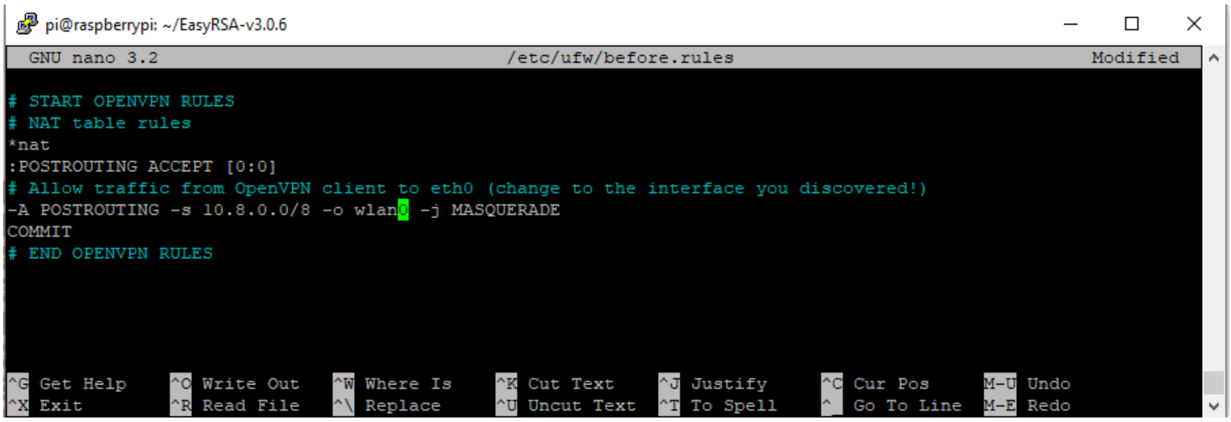
Type **sudo apt install ufw** install the **ufw** package that it is related to the firewall. After it done, open the file before.rules with the nano editor to change the configuration by inserting **sudo nano /etc/ufw/before.rules**



```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo nano /etc/ufw/before.rules
pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

Inside the **before.rules** files find the following lines and change the interface eth0 to the one that was shown on the previous step. Those lines are responsible to masquerade any traffic coming from the OpenVPN (Ellingwood and Drake, 2019).

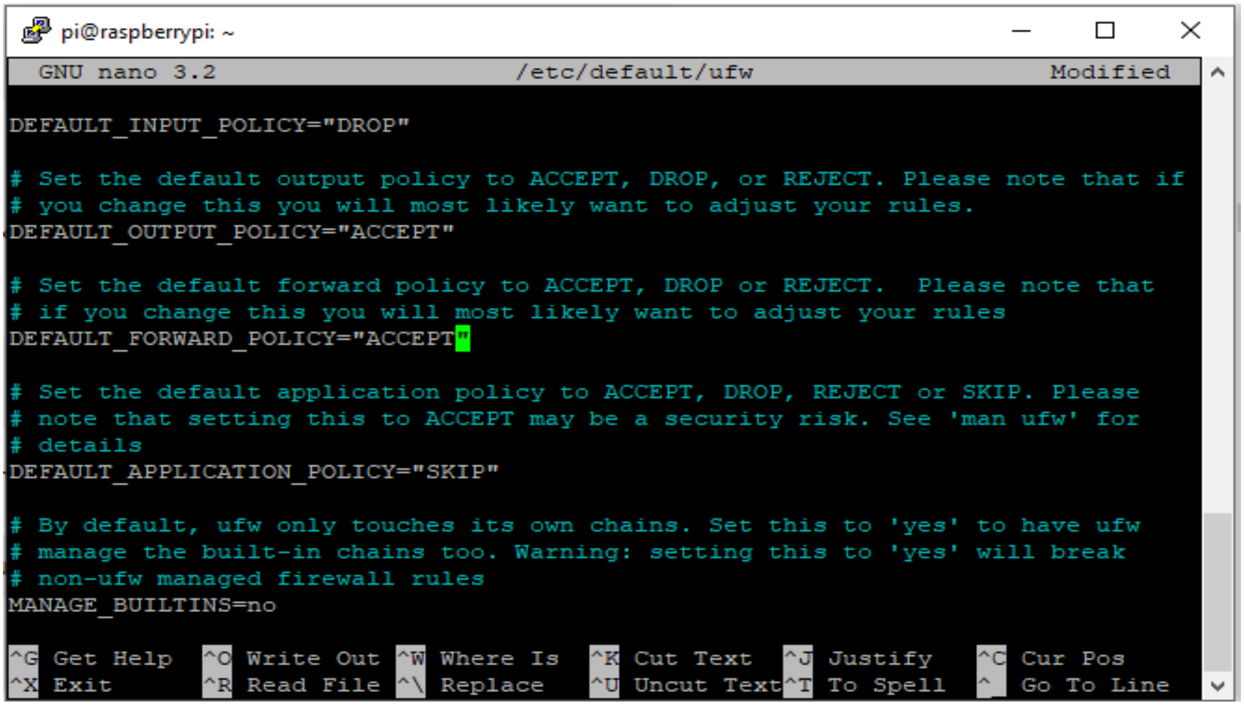
```
# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to eth0 (change to the interface you discovered!)
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
COMMIT
# END OPENVPN RULES
```



```
# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to eth0 (change to the interface you discovered!)
-A POSTROUTING -s 10.8.0.0/8 -o wlan0 -j MASQUERADE
COMMIT
# END OPENVPN RULES
```

Save the file (Ctrl + X) and exit.

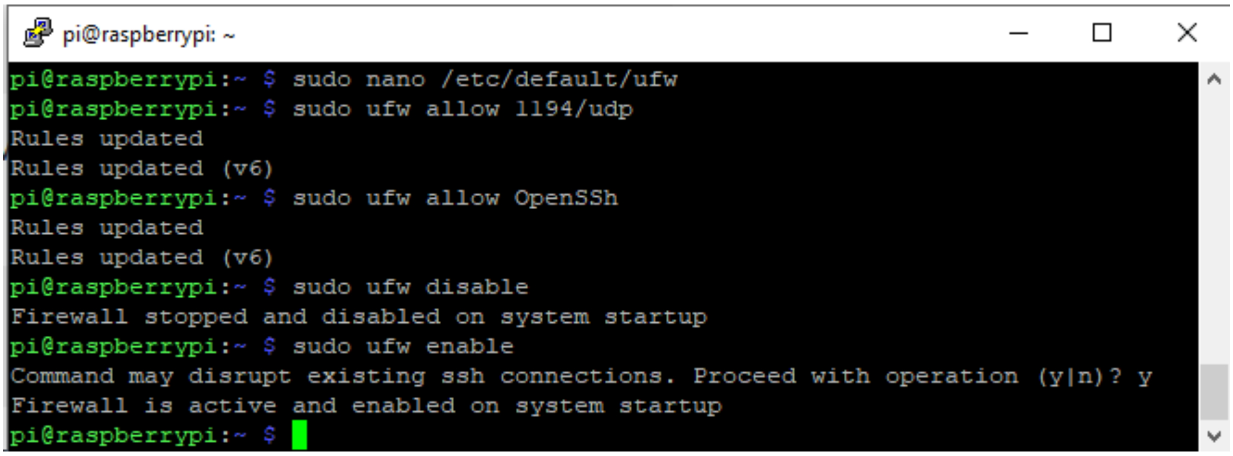
Open the **/etc/default/ufw** file by typing **sudo nano /etc/default/ufw** to set the UFW to allow forwarded packets. Once inside the file find the line DEFAULT_FORWARD_POLICY and change the value “DROP” to “ACCEPT”.



```
DEFAULT_INPUT_POLICY="DROP"

# Set the default output policy to ACCEPT, DROP, or REJECT. Please note that if
# you change this you will most likely want to adjust your rules.
DEFAULT_OUTPUT_POLICY="ACCEPT"

# Set the default forward policy to ACCEPT, DROP or REJECT.  Please note that
# if you change this you will most likely want to adjust your rules
DEFAULT_FORWARD_POLICY="ACCEPT"

# Set the default application policy to ACCEPT, DROP, REJECT or SKIP. Please
# note that setting this to ACCEPT may be a security risk. See 'man ufw' for
# details
DEFAULT_APPLICATION_POLICY="SKIP"

# By default, ufw only touches its own chains. Set this to 'yes' to have ufw
# manage the built-in chains too. Warning: setting this to 'yes' will break
# non-ufw managed firewall rules
MANAGE_BUILTINS=no
```

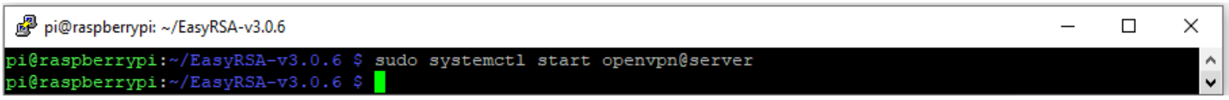
After saving and closing the file, type the commands **sudo ufw allow 1194/udp** and then **sudo ufw allow OpenSSH** to configure the firewall to allow traffic from **OpenVPN** and **SSH** traffic.

Type the commands **sudo ufw disable** and **sudo ufw enable** to make sure the previous configuration was placed.



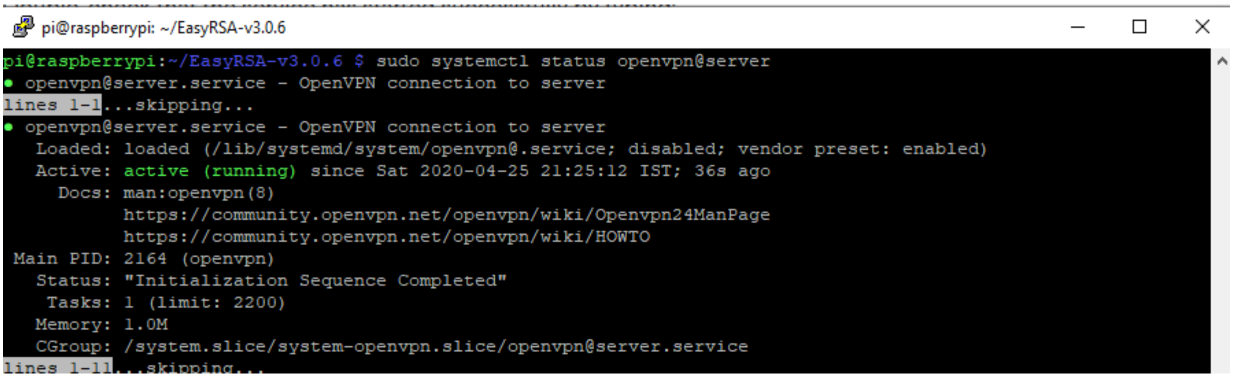
```
pi@raspberrypi:~ $ sudo nano /etc/default/ufw
pi@raspberrypi:~ $ sudo ufw allow 1194/udp
Rules updated
Rules updated (v6)
pi@raspberrypi:~ $ sudo ufw allow OpenSSh
Rules updated
Rules updated (v6)
pi@raspberrypi:~ $ sudo ufw disable
Firewall stopped and disabled on system startup
pi@raspberrypi:~ $ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
pi@raspberrypi:~ $
```

Type the command **sudo systemctl start openvpn@server** to start the server in the raspberry pi.



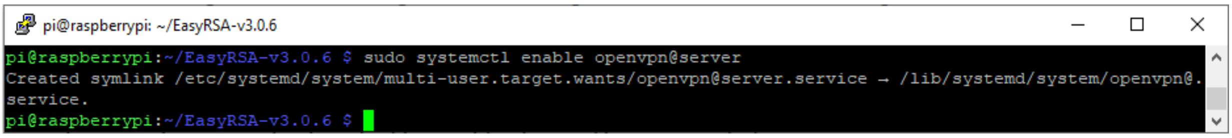
```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo systemctl start openvpn@server
pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

If the server is working correctly, the output looks like this:



```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo systemctl status openvpn@server
● openvpn@server.service - OpenVPN connection to server
lines 1-1...skipping...
● openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-04-25 21:25:12 IST; 36s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
 Main PID: 2164 (openvpn)
   Status: "Initialization Sequence Completed"
    Tasks: 1 (limit: 2200)
   Memory: 1.0M
   CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
lines 1-11...skipping...
```

Type the command **sudo systemctl enable openvpn@server** to enable the server to start to run automatically when Raspbian boots.



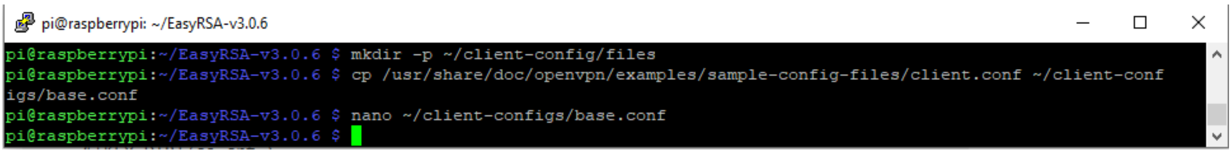
```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo systemctl enable openvpn@server
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn@server.service → /lib/systemd/system/openvpn@.
service.
pi@raspberrypi:~/EasyRSA-v3.0.6 $ 
```

Client configuration

The following steps are going to create a client configuration infrastructure and script that can be run to create client files easily without the necessity of typing multiples commands to configure each client.

Firstly, create a directory to store the files for the configuration of each client by typing **mkdir -p ~/client-configs/files** and copy an example of the client configuration file into the new directory.

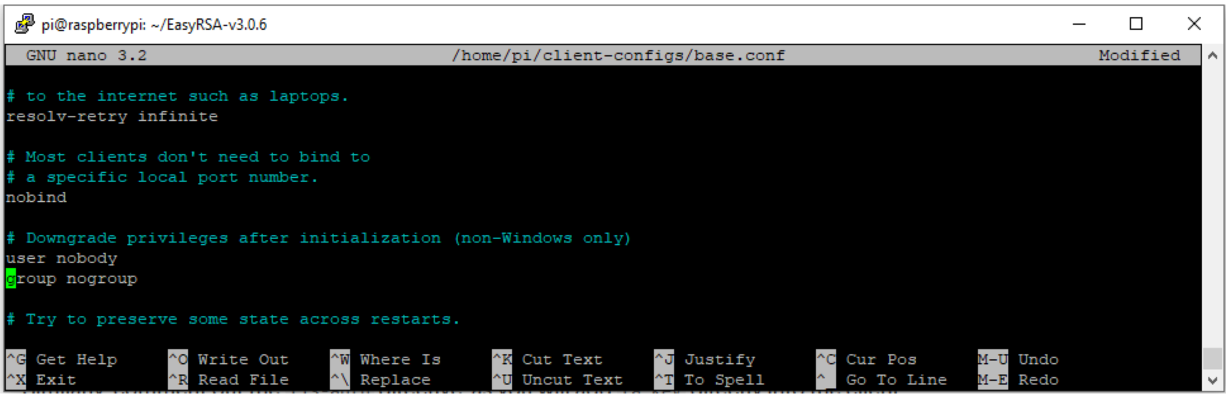
Type **cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-configs/base.conf**. The file needs to be edited, for that, type: **nano ~/client-configs/base.conf**



```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ mkdir -p ~/client-config/files
pi@raspberrypi:~/EasyRSA-v3.0.6 $ cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-conf
igs/base.conf
pi@raspberrypi:~/EasyRSA-v3.0.6 $ nano ~/client-configs/base.conf
pi@raspberrypi:~/EasyRSA-v3.0.6 $ 
```

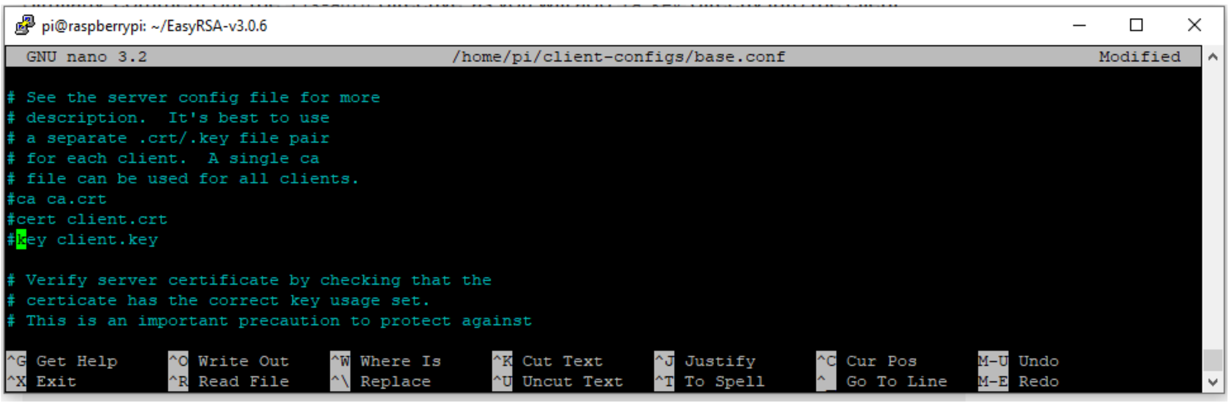
Inside the file, find the line **remote** that points to the client the location of the OpenVPN server. Find the line and add the **public IP address** of the OpenVPN server.

Find the line about protocols and check if **proto UDP** is uncommented and go to line of user and groups to uncomment them as well.



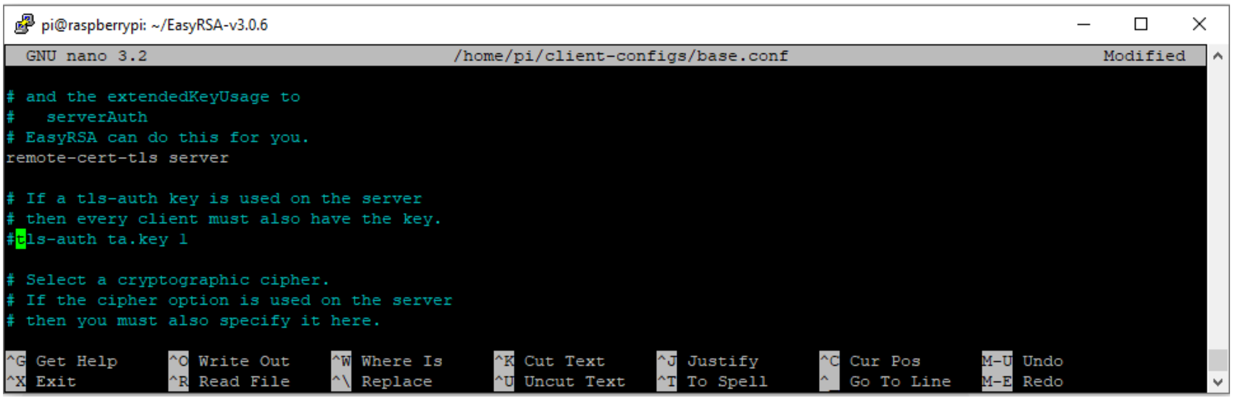
```
  GNU nano 3.2                    /home/pi/client-configs/base.conf                    Modified

# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows only)
user nobody
group nogroup

# Try to preserve some state across restarts.

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line M-E Redo
```

Comment the lines for **ca ca.crt**, cert **client.crt** and **key client.key.**



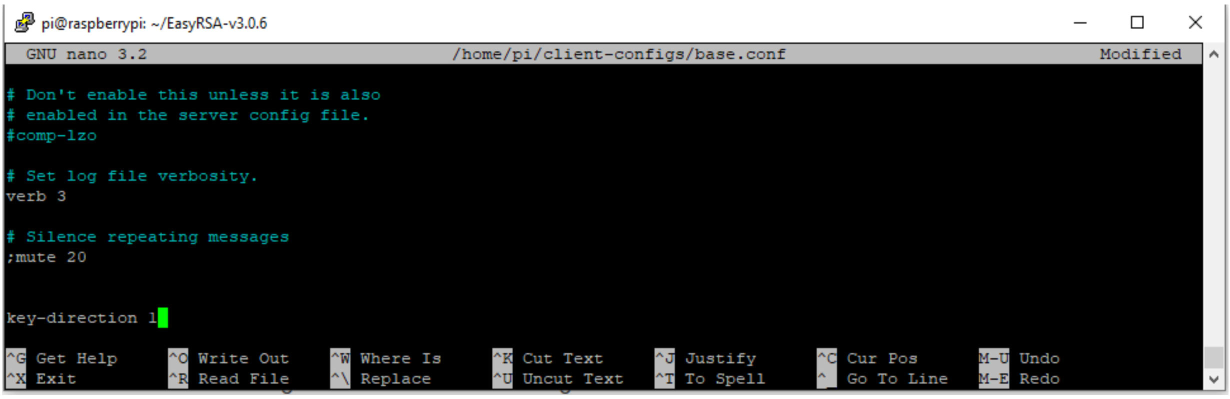
```
  GNU nano 3.2                 /home/pi/client-configs/base.conf                 Modified

# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
#ca ca.crt
#cert client.crt
#key client.key

# Verify server certificate by checking that the
# certicate has the correct key usage set.
# This is an important precaution to protect against
```

Also, comment the line **tls-auth ta.key 1**, because the **ta.key** is going to be added to the to the client configuration file.



```
  GNU nano 3.2                 /home/pi/client-configs/base.conf                 Modified

# and the extendedKeyUsage to
#   serverAuth
# EasyRSA can do this for you.
remote-cert-tls server

# If a tls-auth key is used on the server
# then every client must also have the key.
#tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
```

Add the line **key-direction 1** to the file. It has to be set to 1 in order to the VPN works correctly.



```
  GNU nano 3.2                 /home/pi/client-configs/base.conf                 Modified

# Don't enable this unless it is also
# enabled in the server config file.
#comp-lzo

# Set log file verbosity.
verb 3

# Silence repeating messages
;mute 20


key-direction 1
```

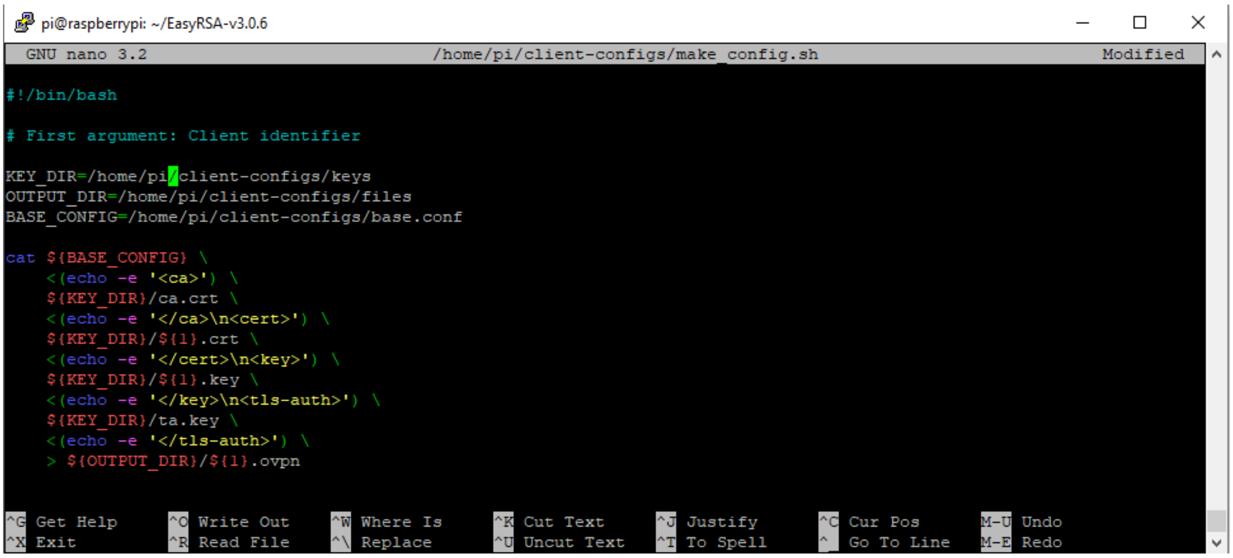
Save the configurations and close the file.

The next step is to create a script inside the **~/client-configs** directory that puts together the information between the configuration file to the certificates files, keys and encryption. Type **nano ~/client-configs/make_config.sh**

Inside the file, add the following lines. This script creates a client **ovpn** file by concatenating the files: the server configuration file and the keys and certificates created in the previous steps. The output of the script is the creating of a file called **client1.ovpn** that is used to connect the clients to the OpenVPN server.

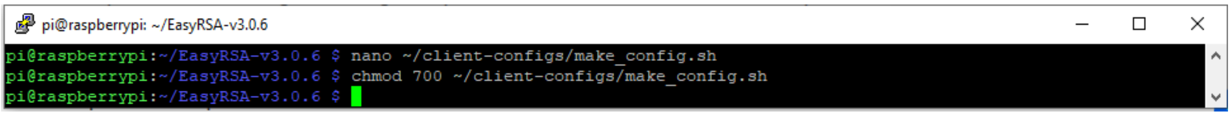
```
#!/bin/bash

# First argument: Client identifier

KEY_DIR=/home/pi/client-configs/keys
OUTPUT_DIR=/home/pi/client-configs/files
BASE_CONFIG=/home/pi/client-configs/base.conf

cat ${BASE_CONFIG} \
    <(echo -e '<ca>') \
    ${KEY_DIR}/ca.crt \
    <(echo -e '</ca>\n<cert>') \
    ${KEY_DIR}/${1}.crt \
    <(echo -e '</cert>\n<key>') \
    ${KEY_DIR}/${1}.key \
    <(echo -e '</key>\n<tls-auth>') \
    ${KEY_DIR}/ta.key \
    <(echo -e '</tls-auth>') \
    > ${OUTPUT_DIR}/${1}.ovpn
```

Change the path to be according to your non-user account. Save the file (Cttl + X) and exit it.



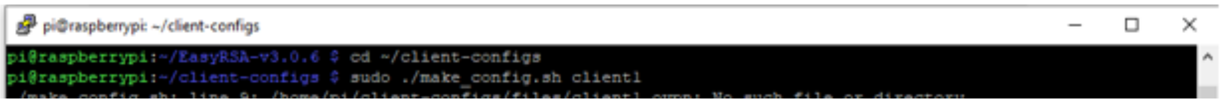
```
  GNU nano 3.2                    /home/pi/client-configs/make_config.sh                    Modified

#!/bin/bash

# First argument: Client identifier

KEY_DIR=/home/pi/client-configs/keys
OUTPUT_DIR=/home/pi/client-configs/files
BASE_CONFIG=/home/pi/client-configs/base.conf

cat ${BASE_CONFIG} \
    <(echo -e '<ca>') \
    ${KEY_DIR}/ca.crt \
    <(echo -e '</ca>\n<cert>') \
    ${KEY_DIR}/${1}.crt \
    <(echo -e '</cert>\n<key>') \
    ${KEY_DIR}/${1}.key \
    <(echo -e '</key>\n<tls-auth>') \
    ${KEY_DIR}/ta.key \
    <(echo -e '</tls-auth>') \
    > ${OUTPUT_DIR}/${1}.ovpn

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  M-E Redo
```

Change the permissions of the file making sure the file is marking as executable by typing **chmod 700 ~/client-configs/make_config.sh**



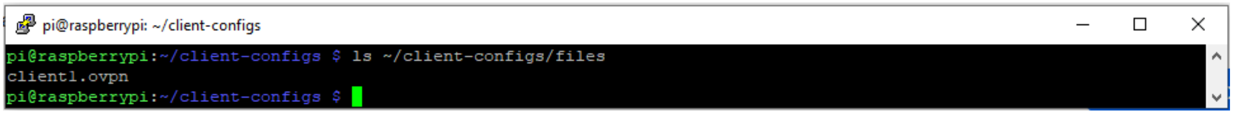
```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ nano ~/client-configs/make_config.sh
pi@raspberrypi:~/EasyRSA-v3.0.6 $ chmod 700 ~/client-configs/make_config.sh
pi@raspberrypi:~/EasyRSA-v3.0.6 $ 
```

Type the command **cd ~/client-configs** to go to the ~/client-configs directory and type the command **sudo ./make_config.sh client1** to run the script and generate a client1.ovpn file for the client1 that was configured on the section d.



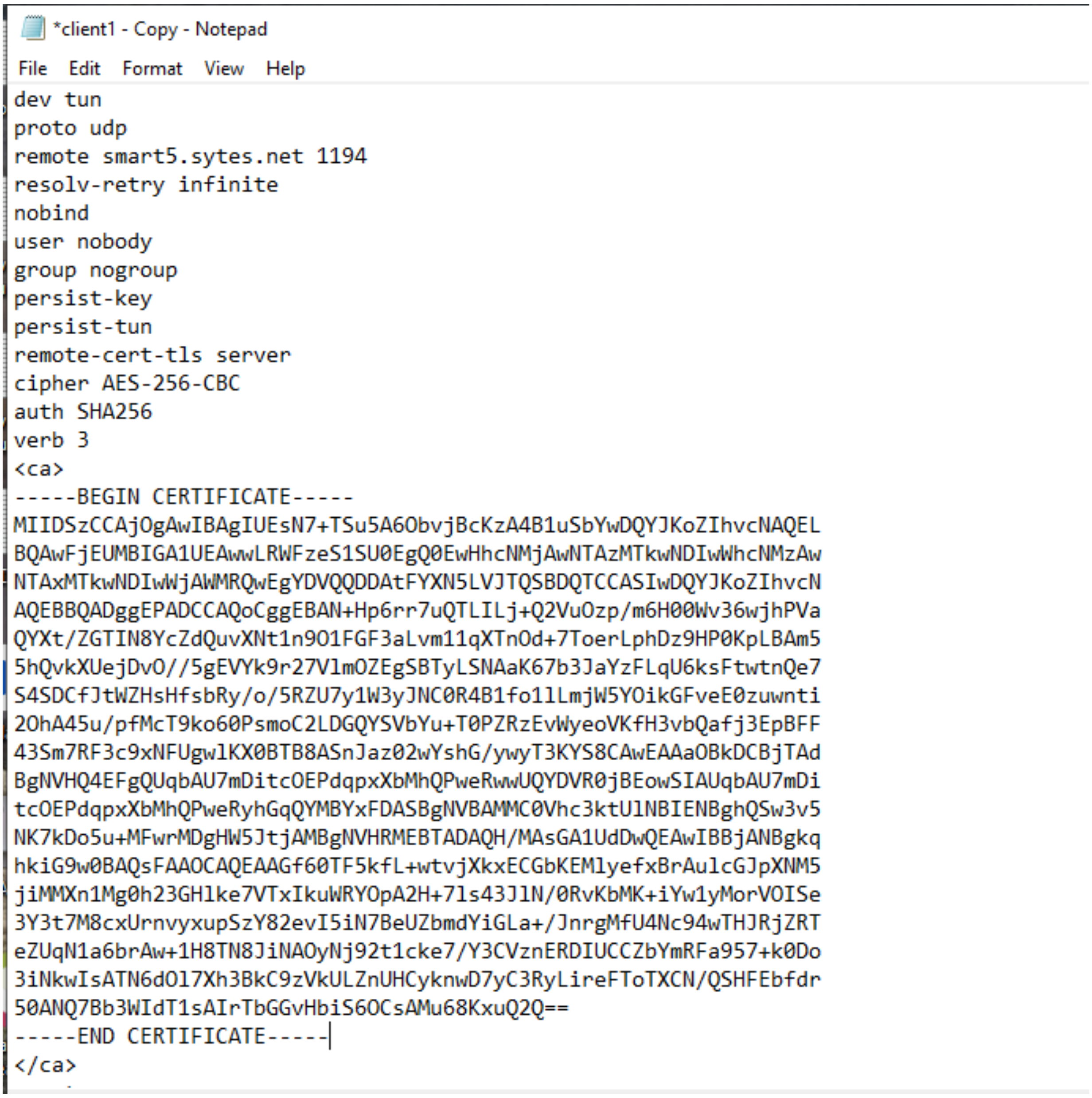
```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ cd ~/client-configs
pi@raspberrypi:~/client-configs $ sudo ./make_config.sh client1
./make_config.sh: line 8: /home/pi/client-configs/files/client1.ovpn: No such file or directory
```

Type the command **ls ~/client-configs/files** to confirm that the file was created. The output should be **client1.ovpn.**

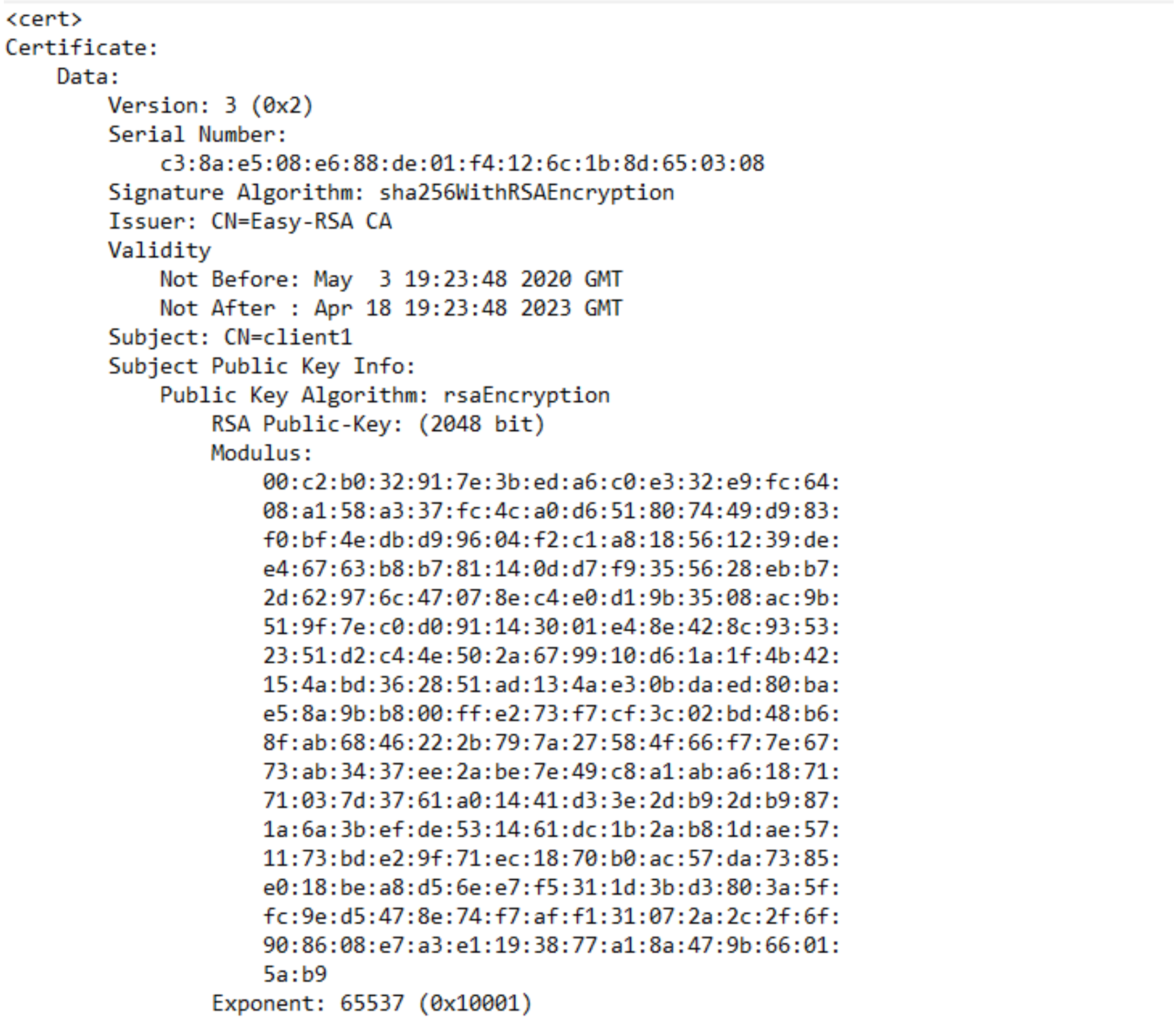


```
pi@raspberrypi:~/client-configs $ ls ~/client-configs/files
client1.ovpn
pi@raspberrypi:~/client-configs $ 
```

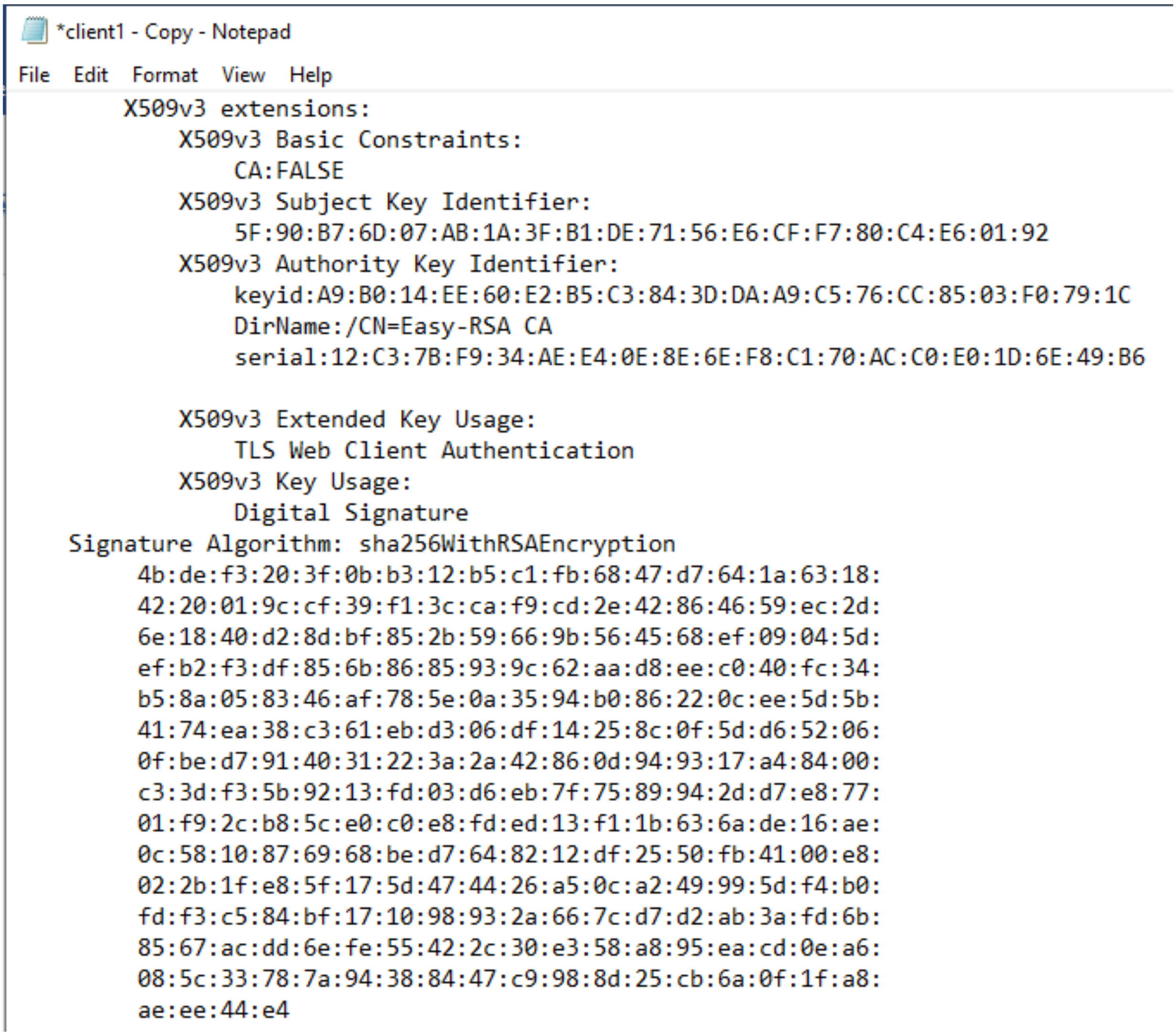
Those configurations are going to generate a client file that look like this:



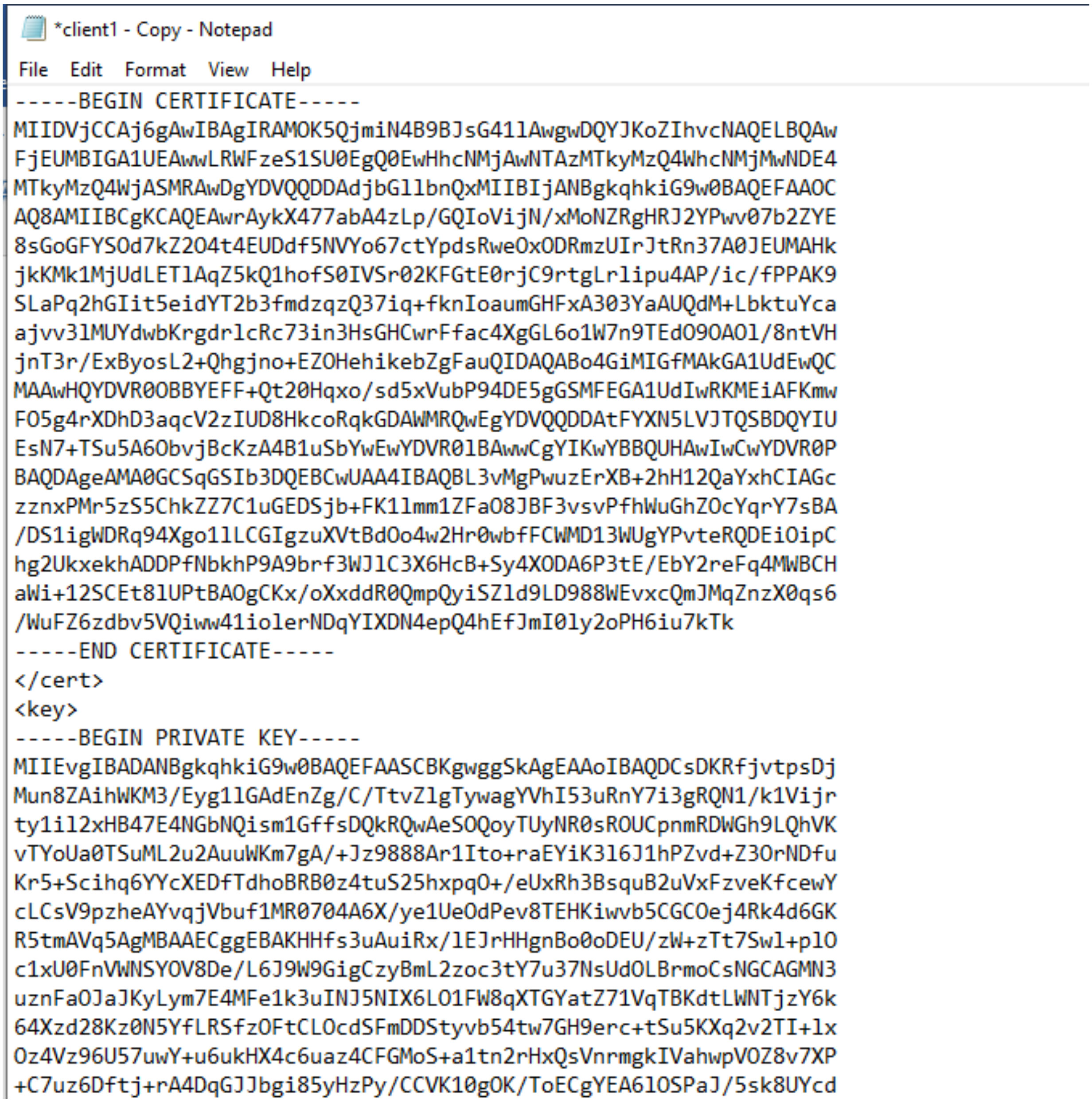
```
dev tun
proto udp
remote smart5.sytes.net 1194
resolv-retry infinite
nobind
user nobody
group nogroup
persist-key
persist-tun
remote-cert-tls server
cipher AES-256-CBC
auth SHA256
verb 3
<ca>
-----BEGIN CERTIFICATE-----
MIIDSzCCAjOgAwIBAgIUEsN7+TSu5A6ObvjBcKzA4B1uSbYwDQYJKoZIhvcNAQEL
BQAwFjEUMBIGA1UEAwwLRWFzeS1SU0EgQ0EwHhcNMjAwNTAzMTkwNDIwWhcNMzAw
NTAxMTkwNDIwWjAWMRQwEgYDVQQDDAtFYXN5LVJTQSBDQTCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAN+Hp6rr7uQTLILj+Q2VuOzp/m6H00Wv36wjhPVa
QYXt/ZGTIN8YcZdQuvXNt1n9O1FGF3aLvm11qXTnOd+7ToerLphDz9HP0KpLBAm5
5hQvkXUejDvO//5gEVYk9r27VlmOZEgSBTyLSNAaK67b3JaYzFLqU6ksFtwtnQe7
S4SDCfJtWZHsHfsbRy/o/5RZU7y1W3yJNC0R4B1fo1lLmjW5YOikGFveE0zuwnti
2OhA45u/pfMcT9ko60PsmoC2LDGQYSVbYu+T0PZRzEvWyeoVKfH3vbQafj3EpBFF
43Sm7RF3c9xNFUgwlKX0BTB8ASnJaz02wYshG/ywyT3KYS8CAwEAAaOBkDCBjTAd
BgNVHQ4EFgQUqbAU7mDitcOEPdqpxXbMhQPweRwwUQYDVR0jBEowSIAUqbAU7mDi
tcOEPdqpxXbMhQPweRyhGqQYMBYxFDASBgNVBAMMC0Vhc3ktUlNBIENBghQSw3v5
NK7kDo5u+MFwrMDgHW5JtjAMBgNVHRMEBTADAQH/MAsGA1UdDwQEAwIBBjANBgkq
hkiG9w0BAQsFAAOCAQEAAGf60TF5kfL+wtvjXkxECGbKEMlyefxBrAulcGJpXNM5
jiMMXn1Mg0h23GHlke7VTxIkuWRYOpA2H+7ls43JlN/0RvKbMK+iYw1yMorVOISe
3Y3t7M8cxUrnvyxupSzY82evI5iN7BeUZbmdYiGLa+/JnrgMfU4Nc94wTHJRjZRT
eZUqN1a6brAw+1H8TN8JiNAOyNj92t1cke7/Y3CVznERDIUCCZbYmRFa957+k0Do
3iNkwIsATN6dOl7Xh3BkC9zVkULZnUHCyknwD7yC3RyLireFToTXCN/QSHFEbfdr
50ANQ7Bb3WIdT1sAIrTbGGvHbiS6OCsAMu68KxuQ2Q==
-----END CERTIFICATE-----
</ca>
```



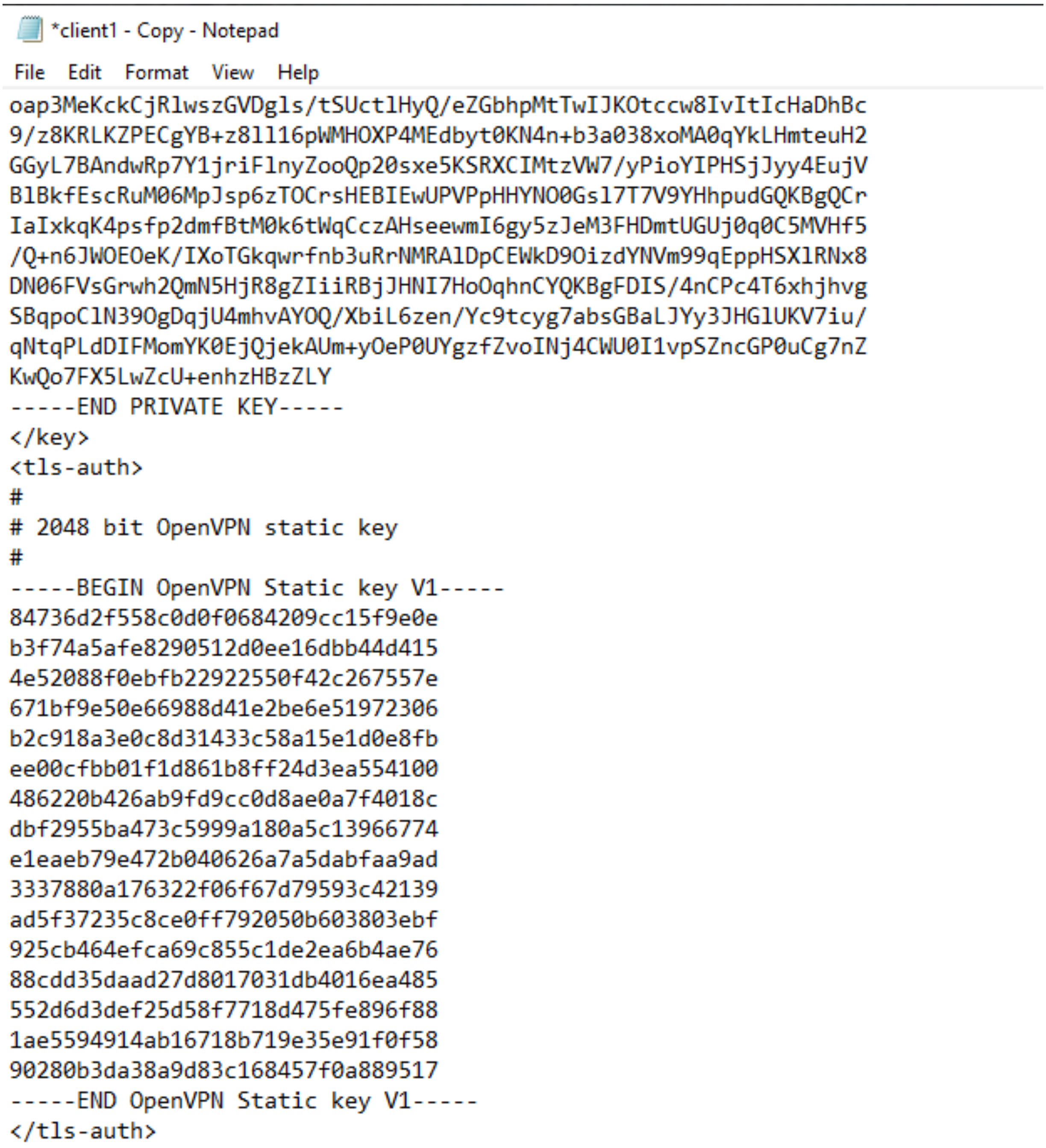
```
<cert>
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            c3:8a:e5:08:e6:88:de:01:f4:12:6c:1b:8d:65:03:08
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=Easy-RSA CA
        Validity
            Not Before: May  3 19:23:48 2020 GMT
            Not After : Apr 18 19:23:48 2023 GMT
        Subject: CN=client1
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:c2:b0:32:91:7e:3b:ed:a6:c0:e3:32:e9:fc:64:
                    08:a1:58:a3:37:fc:4c:a0:d6:51:80:74:49:d9:83:
                    f0:bf:4e:db:d9:96:04:f2:c1:a8:18:56:12:39:de:
                    e4:67:63:b8:b7:81:14:0d:d7:f9:35:56:28:eb:b7:
                    2d:62:97:6c:47:07:8e:c4:e0:d1:9b:35:08:ac:9b:
                    51:9f:7e:c0:d0:91:14:30:01:e4:8e:42:8c:93:53:
                    23:51:d2:c4:4e:50:2a:67:99:10:d6:1a:1f:4b:42:
                    15:4a:bd:36:28:51:ad:13:4a:e3:0b:da:ed:80:ba:
                    e5:8a:9b:b8:00:ff:e2:73:f7:cf:3c:02:bd:48:b6:
                    8f:ab:68:46:22:2b:79:7a:27:58:4f:66:f7:7e:67:
                    73:ab:34:37:ee:2a:be:7e:49:c8:a1:ab:a6:18:71:
                    71:03:7d:37:61:a0:14:41:d3:3e:2d:b9:2d:b9:87:
                    1a:6a:3b:ef:de:53:14:61:dc:1b:2a:b8:1d:ae:57:
                    11:73:bd:e2:9f:71:ec:18:70:b0:ac:57:da:73:85:
                    e0:18:be:a8:d5:6e:e7:f5:31:1d:3b:d3:80:3a:5f:
                    fc:9e:d5:47:8e:74:f7:af:f1:31:07:2a:2c:2f:6f:
                    90:86:08:e7:a3:e1:19:38:77:a1:8a:47:9b:66:01:
                    5a:b9
                Exponent: 65537 (0x10001)
```



```
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                5F:90:B7:6D:07:AB:1A:3F:B1:DE:71:56:E6:CF:F7:80:C4:E6:01:92
            X509v3 Authority Key Identifier:
                keyid:A9:B0:14:EE:60:E2:B5:C3:84:3D:DA:A9:C5:76:CC:85:03:F0:79:1C
                DirName:/CN=Easy-RSA CA
                serial:12:C3:7B:F9:34:AE:E4:0E:8E:6E:F8:C1:70:AC:C0:E0:1D:6E:49:B6

            X509v3 Extended Key Usage:
                TLS Web Client Authentication
            X509v3 Key Usage:
                Digital Signature
    Signature Algorithm: sha256WithRSAEncryption
         4b:de:f3:20:3f:0b:b3:12:b5:c1:fb:68:47:d7:64:1a:63:18:
         42:20:01:9c:cf:39:f1:3c:ca:f9:cd:2e:42:86:46:59:ec:2d:
         6e:18:40:d2:8d:bf:85:2b:59:66:9b:56:45:68:ef:09:04:5d:
         ef:b2:f3:df:85:6b:86:85:93:9c:62:aa:d8:ee:c0:40:fc:34:
         b5:8a:05:83:46:af:78:5e:0a:35:94:b0:86:22:0c:ee:5d:5b:
         41:74:ea:38:c3:61:eb:d3:06:df:14:25:8c:0f:5d:d6:52:06:
         0f:be:d7:91:40:31:22:3a:2a:42:86:0d:94:93:17:a4:84:00:
         c3:3d:f3:5b:92:13:fd:03:d6:eb:7f:75:89:94:2d:d7:e8:77:
         01:f9:2c:b8:5c:e0:c0:e8:fd:ed:13:f1:1b:63:6a:de:16:ae:
         0c:58:10:87:69:68:be:d7:64:82:12:df:25:50:fb:41:00:e8:
         02:2b:1f:e8:5f:17:5d:47:44:26:a5:0c:a2:49:99:5d:f4:b0:
         fd:f3:c5:84:bf:17:10:98:93:2a:66:7c:d7:d2:ab:3a:fd:6b:
         85:67:ac:dd:6e:fe:55:42:2c:30:e3:58:a8:95:ea:cd:0e:a6:
         08:5c:33:78:7a:94:38:84:47:c9:98:8d:25:cb:6a:0f:1f:a8:
         ae:ee:44:e4
```



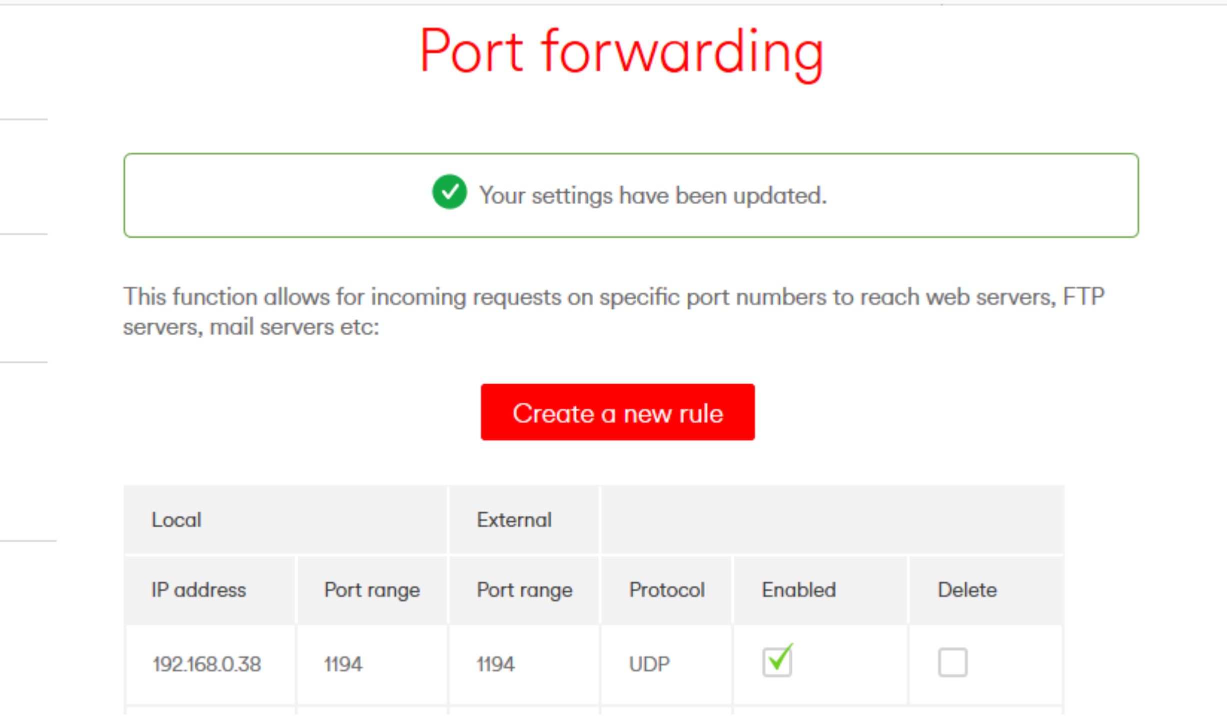
```
-----BEGIN CERTIFICATE-----
MIIDVjCCAj6gAwIBAgIRAMOK5QjmiN4B9BJsG41lAwgwDQYJKoZIhvcNAQELBQAw
FjEUMBIGA1UEAwwLRWFzeS1SU0EgQ0EwHhcNMjAwNTAzMTkyMzQ4WhcNMjMwNDE4
MTkyMzQ4WjASMRAwDgYDVQQDDAdjbGllbnQxMIIBIjANBgkqhkiG9w0BAQEFAAOC
AQ8AMIIBCgKCAQEAwrAykX477abA4zLp/GQIoVijN/xMoNZRgHRJ2YPwv07b2ZYE
8sGoGFYSOd7kZ2O4t4EUDdf5NVYo67ctYpdsRweOxODRmzUIrJtRn37A0JEUMAHk
jkKMk1MjUdLETlAqZ5kQ1hofS0IVSr02KFGtE0rjC9rtgLrlipu4AP/ic/fPPAK9
SLaPq2hGIit5eidYT2b3fmdzqzQ37iq+fknIoaumGHFxA303YaAUQdM+LbktuYca
ajvv3lMUYdwbKrgdrlcRc73in3HsGHCwrFfac4XgGL6o1W7n9TEdO9OAOl/8ntVH
jnT3r/ExByosL2+Qhgjno+EZOHehikebZgFauQIDAQABo4GiMIGfMAkGA1UdEwQC
MAAwHQYDVR0OBBYEFF+Qt20Hqxo/sd5xVubP94DE5gGSMFEGA1UdIwRKMEiAFKmw
FO5g4rXDhD3aqcV2zIUD8HkcoRqkGDAWMRQwEgYDVQQDDAtFYXN5LVJTQSBDQYIU
EsN7+TSu5A6ObvjBcKzA4B1uSbYwEwYDVR0lBAwwCgYIKwYBBQUHAwIwCwYDVR0P
BAQDAgeAMA0GCSqGSIb3DQEBCwUAA4IBAQBL3vMgPwuzErXB+2hH12QaYxhCIAGc
zznxPMr5zS5ChkZZ7C1uGEDSjb+FK1lmm1ZFaO8JBF3vsvPfhWuGhZOcYqrY7sBA
/DS1igWDRq94Xgo1lLCGIgzuXVtBdOo4w2Hr0wbfFCWMD13WUgYPvteRQDEiOipC
hg2UkxekhADDPfNbkhP9A9brf3WJlC3X6HcB+Sy4XODA6P3tE/EbY2reFq4MWBCH
aWi+12SCEt8lUPtBAOgCKx/oXxddR0QmpQyiSZld9LD988WEvxcQmJMqZnzX0qs6
/WuFZ6zdbv5VQiww41iolerNDqYIXDN4epQ4hEfJmI0ly2oPH6iu7kTk
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDCsDKRfjvtpsDj
Mun8ZAihWKM3/Eyg1lGAdEnZg/C/TtvZlgTywagYVhI53uRnY7i3gRQN1/k1Vijr
ty1il2xHB47E4NGbNQism1GffsDQkRQwAeSOQoyTUyNR0sROUCpnmRDWGh9LQhVK
vTYoUa0TSuML2u2AuuWKm7gA/+Jz9888Ar1Ito+raEYiK3l6J1hPZvd+Z3OrNDfu
Kr5+Scihq6YYcXEDfTdhoBRB0z4tuS25hxpqO+/eUxRh3BsquB2uVxFzveKfcewY
cLCsV9pzheAYvqjVbuf1MR0704A6X/ye1UeOdPev8TEHKiwvb5CGCOej4Rk4d6GK
R5tmAVq5AgMBAAECggEBAKHHfs3uAuiRx/lEJrHHgnBo0oDEU/zW+zTt7Swl+plO
c1xU0FnVWNSYOV8De/L6J9W9GigCzyBmL2zoc3tY7u37NsUdOLBrmoCsNGCAGMN3
uznFaOJaJKyLym7E4MFe1k3uINJ5NIX6LO1FW8qXTGYatZ71VqTBKdtLWNTjzY6k
64Xzd28Kz0N5YfLRSfzOFtCLOcdSFmDDStyvb54tw7GH9erc+tSu5KXq2v2TI+lx
Oz4Vz96U57uwY+u6ukHX4c6uaz4CFGMoS+a1tn2rHxQsVnrmgkIVahwpVOZ8v7XP
+C7uz6Dftj+rA4DqGJJbgi85yHzPy/CCVK10gOK/ToECgYEA6lOSPaJ/5sk8UYcd
```



```
oap3MeKckCjRlwszGVDgls/tSUctlHyQ/eZGbhpMtTwIJKOtccw8IvItIcHaDhBc
9/z8KRLKZPECgYB+z81l16pWMHOXP4MEdbyt0KN4n+b3a038xoMA0qYkLHmteuH2
GGyL7BAndwRp7Y1jriFlnyZooQp20sxe5KSRXCIMtzVW7/yPioYIPHSjJyy4EujV
BlBkfEscRuM06MpJsp6zTOCrsHEBIEwUPVPpHHYNO0Gsl7T7V9YHhpudGQKBgQCr
IaIxkqK4psfp2dmfBtM0k6tWqCczAHseewmI6gy5zJeM3FHDmtUGUj0q0C5MVHf5
/Q+n6JWOEOeK/IXoTGkqwrfnb3uRrNMRAlDpCEWkD9OizdYNVm99qEppHSXlRNx8
DN06FVsGrwh2QmN5HjR8gZIiiRBjJHNI7HoOqhnCYQKBgFDIS/4nCPc4T6xhjhvg
SBqpoClN39OgDqjU4mhvAYOQ/XbiL6zen/Yc9tcyg7absGBaLJYy3JHGlUKV7iu/
qNtqPLdDIFMomYK0EjQjekAUm+yOeP0UYgzfZvoINj4CWU0I1vpSZncGP0uCg7nZ
KwQo7FX5LwZcU+enhzHBzZLY
-----END PRIVATE KEY-----
</key>
<tls-auth>
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
84736d2f558c0d0f0684209cc15f9e0e
b3f74a5afe8290512d0ee16dbb44d415
4e52088f0ebfb22922550f42c267557e
671bf9e50e66988d41e2be6e51972306
b2c918a3e0c8d31433c58a15e1d0e8fb
ee00cfbb01f1d861b8ff24d3ea554100
486220b426ab9fd9cc0d8ae0a7f4018c
dbf2955ba473c5999a180a5c13966774
e1eaeb79e472b040626a7a5dabfaa9ad
3337880a176322f06f67d79593c42139
ad5f37235c8ce0ff792050b603803ebf
925cb464efca69c855c1de2ea6b4ae76
88cdd35daad27d8017031db4016ea485
552d6d3def25d58f7718d475fe896f88
1ae5594914ab16718b719e35e91f0f58
90280b3da38a9d83c168457f0a889517
-----END OpenVPN Static key V1-----
</tls-auth>
```

Configuring port forward in the router

Once the configuration for the server and the client is done, the configuration in the router has to be done. For that, the user should log in their router and enable por forward. Each router has a different graphic user interface, but basically the user has to find the option port forwarding which sometimes is inside the security tab. The important thing is to enable a rule that forward the traffic to the Raspberry Pi device [192.168.0.38] through the port 1194.



Port forwarding

✓ Your settings have been updated.

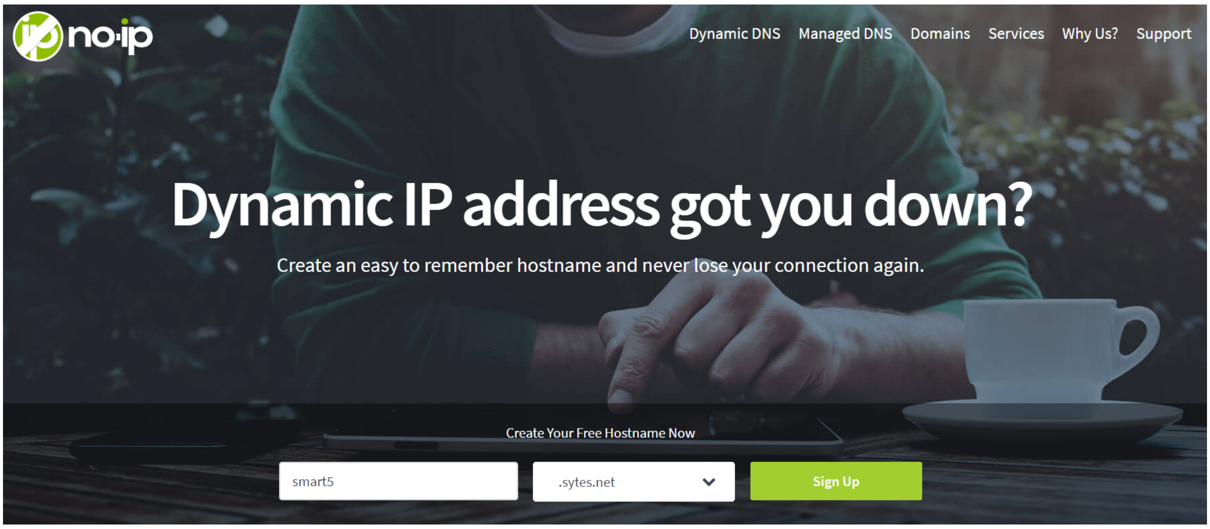
This function allows for incoming requests on specific port numbers to reach web servers, FTP servers, mail servers etc:

Local		External			
IP address	Port range	Port range	Protocol	Enabled	Delete
192.168.0.38	1194	1194	UDP	☑	□

Setting up a free dynamic DNS account with NO-IP

In chapter 3, it was mentioned that dynamic DNS would be needed for this project due to the fact that the public IP address is not static and might change. For this reason, dynamic DNS has to be used in order to make the OpenVPN server always available to the client even with changes in the IP address.

Firstly, access the website https://www.noip.com/ to register a free hostname

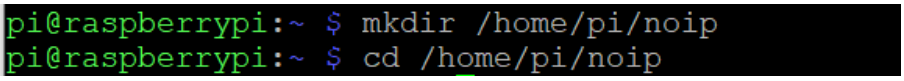

no-ip
Dynamic DNS
Managed DNS
Domains
Services
Why Us?
Support
Dynamic IP address got you down?
Create an easy to remember hostname and never lose your connection again.
Create Your Free Hostname Now
Sign Up

smart5

.sytes.net

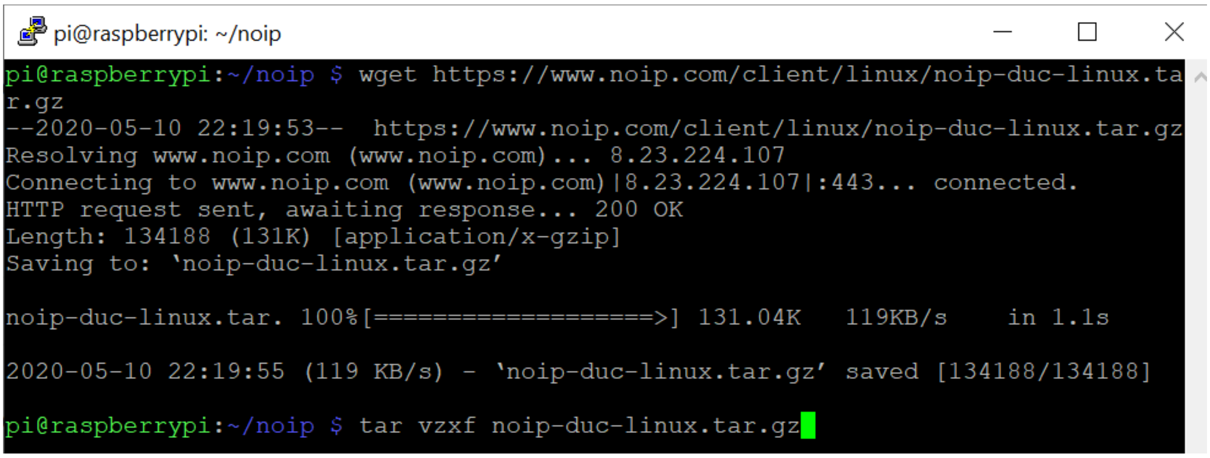
Download and install the Dynamic Update Client (DUC) which is important to keep the hostname updated with the current IP address.

The first step is to create a directory for the client software to be installed by entering the command line **mkdir /home/pi/noip** followed by the command **cd /home/pi/noip.**



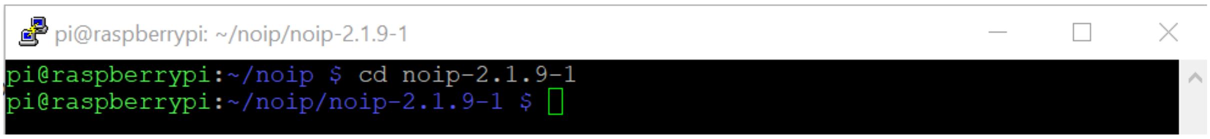
```
pi@raspberrypi:~ $ mkdir /home/pi/noip
pi@raspberrypi:~ $ cd /home/pi/noip
```

Create the folders for DUC and download the software by typing the command **wget https://www.noip.com/client/linux/noip-duc-linux.tar.gz** followed by the command **tar vzxf noip-duc-linux.tar.gz.**



```
pi@raspberrypi:~/noip $ wget https://www.noip.com/client/linux/noip-duc-linux.ta
r.gz
--2020-05-10 22:19:53--  https://www.noip.com/client/linux/noip-duc-linux.tar.gz
Resolving www.noip.com (www.noip.com)... 8.23.224.107
Connecting to www.noip.com (www.noip.com)|8.23.224.107|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 134188 (131K) [application/x-gzip]
Saving to: 'noip-duc-linux.tar.gz'

noip-duc-linux.tar. 100%[==================>] 131.04K   119KB/s    in 1.1s

2020-05-10 22:19:55 (119 KB/s) - 'noip-duc-linux.tar.gz' saved [134188/134188]

pi@raspberrypi:~/noip $ tar vzxf noip-duc-linux.tar.gz
```

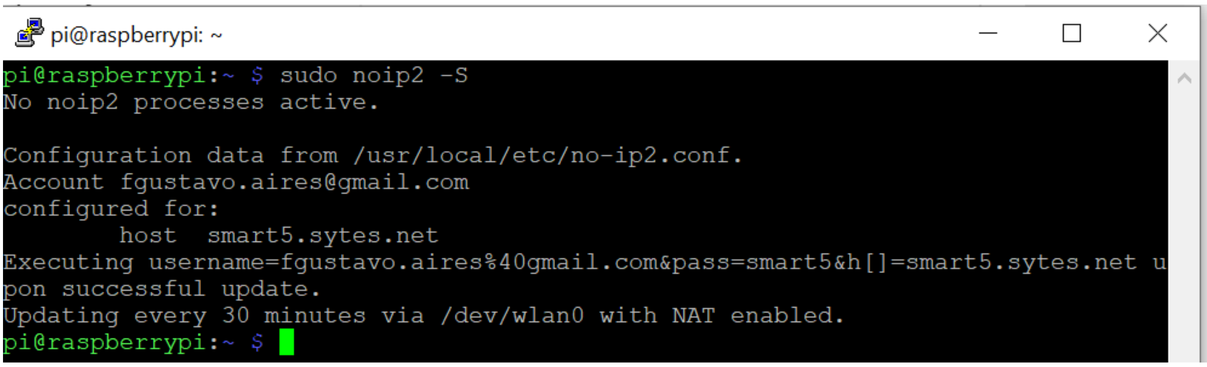
Navigate to the directory where the files are installed by typing the command **cd noip-2.1.9-1.**



```
pi@raspberrypi:~/noip $ cd noip-2.1.9-1
pi@raspberrypi:~/noip/noip-2.1.9-1 $
```

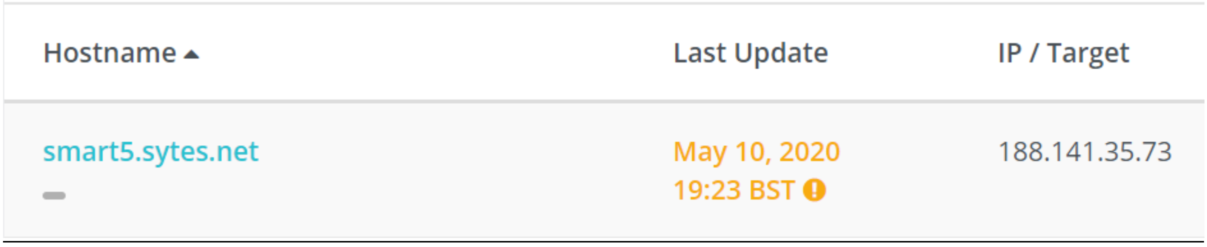
Install the program by typing the commands **sudo make** followed by the command **sudo make install.** After it will be prompted to login with your No-IP account username and password. The next step is reinitializing the raspberry pi.

To confirm the No-IP DUC Client Service is running type the command line **sudo noip2 -S**.



```
pi@raspberrypi:~ $ sudo noip2 -S
No noip2 processes active.

Configuration data from /usr/local/etc/no-ip2.conf.
Account fgustavo.aires@gmail.com
configured for:
        host  smart5.sytes.net
Executing username=fgustavo.aires%40gmail.com&pass=smart5&h[]=smart5.sytes.net u
pon successful update.
Updating every 30 minutes via /dev/wlan0 with NAT enabled.
pi@raspberrypi:~ $
```

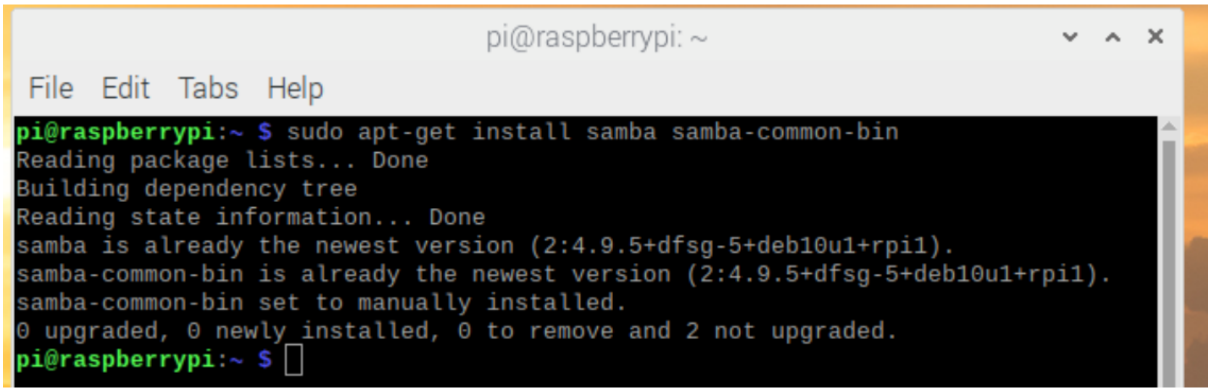
Then accessing No-IP account we can check the concerned DNS has been updated.



Hostname ▲	Last Update	IP / Target
smart5.sytes.net	May 10, 2020 19:23 BST !	188.141.35.73

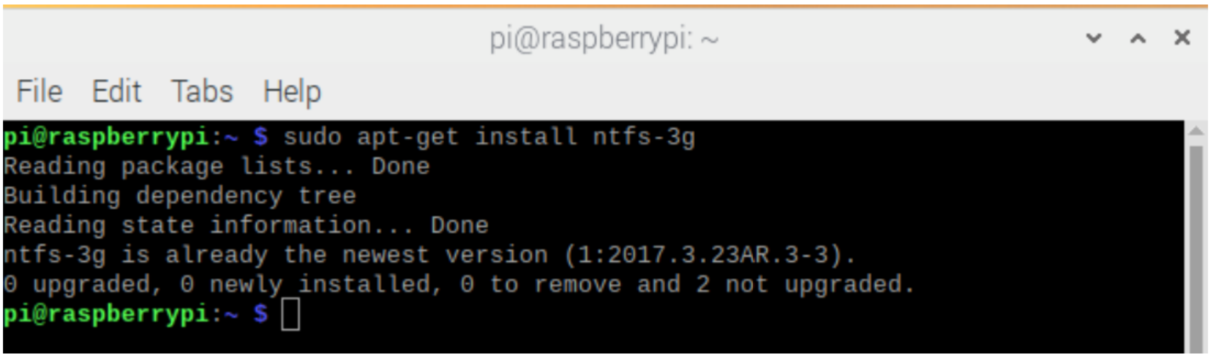
Installing Network Attachment Storage (NAS)

Install the Samba package by typing **Sudo apt-get intall samba samba-common-bin**



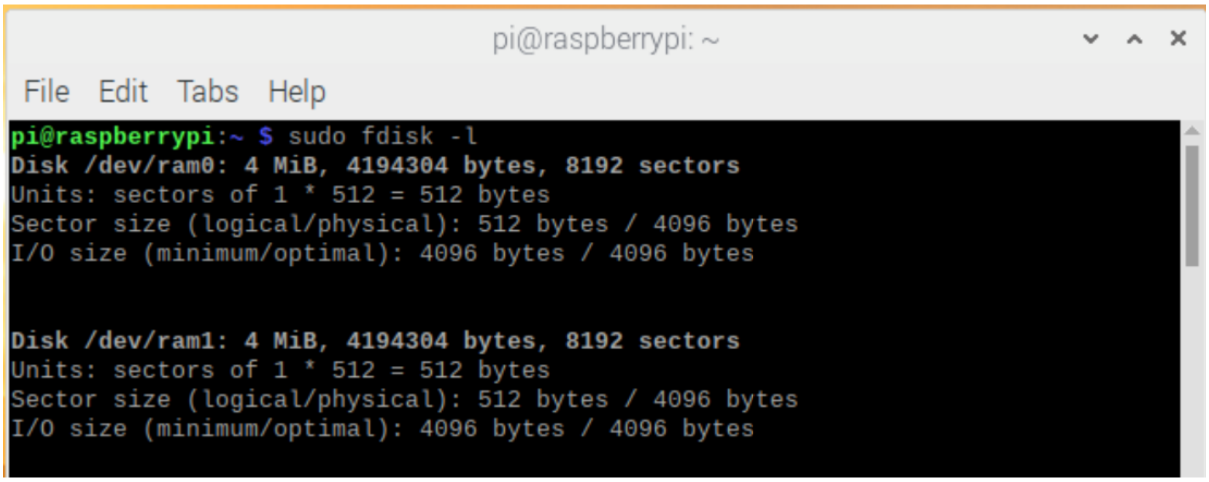
```
pi@raspberrypi:~ $ sudo apt-get install samba samba-common-bin
Reading package lists... Done
Building dependency tree
Reading state information... Done
samba is already the newest version (2:4.9.5+dfsg-5+deb10u1+rpi1).
samba-common-bin is already the newest version (2:4.9.5+dfsg-5+deb10u1+rpi1).
samba-common-bin set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
pi@raspberrypi:~ $
```

Install NTFS-3G by typing **sudo apt-get intall ntfs-3g**



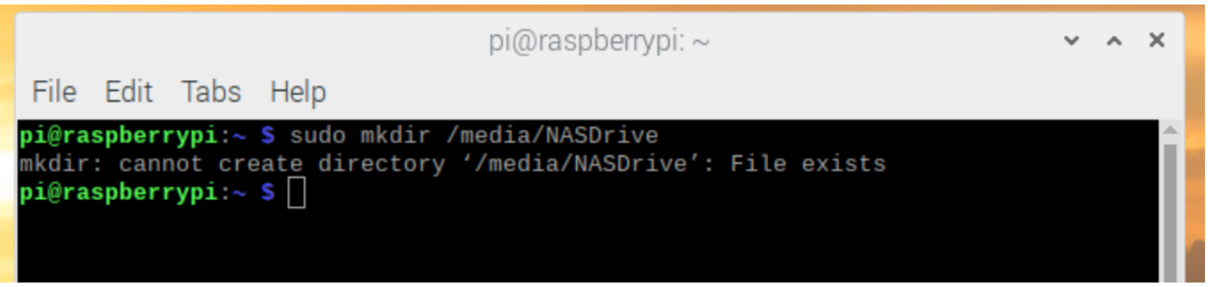
```
pi@raspberrypi:~ $ sudo apt-get install ntfs-3g
Reading package lists... Done
Building dependency tree
Reading state information... Done
ntfs-3g is already the newest version (1:2017.3.23AR.3-3).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
pi@raspberrypi:~ $
```

Verify the location of the USB stick by typing: **sudo dfisk -l**



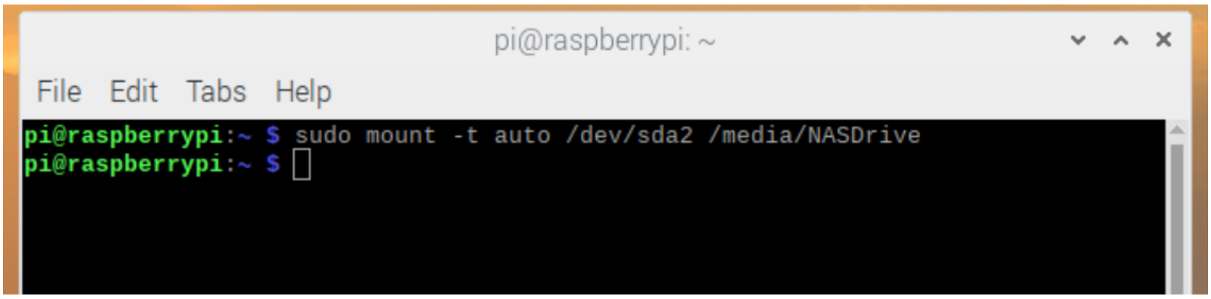
```
pi@raspberrypi:~ $ sudo fdisk -l
Disk /dev/ram0: 4 MiB, 4194304 bytes, 8192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes


Disk /dev/ram1: 4 MiB, 4194304 bytes, 8192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

Create a new file that it will work as NAS: **sudo mkdir /media/NASDrive**



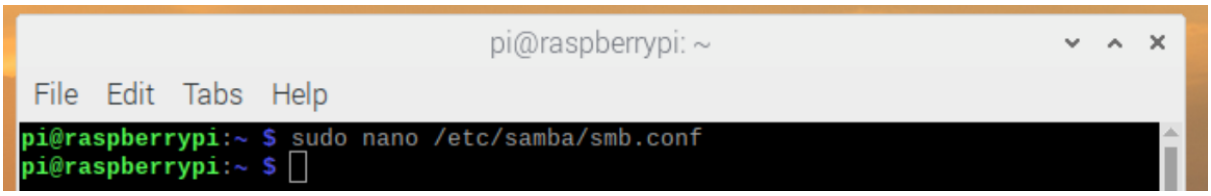
```
pi@raspberrypi:~ $ sudo mkdir /media/NASDrive
mkdir: cannot create directory '/media/NASDrive': File exists
pi@raspberrypi:~ $ 
```

It mounts the drive to the folder: **sudo mount -t auto /dev/sda2 /media/NASDrive**



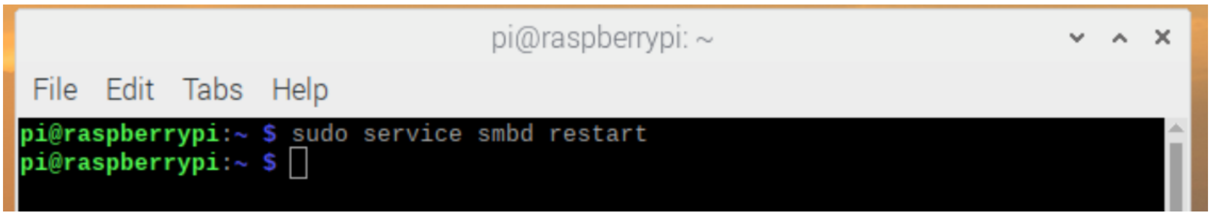
```
pi@raspberrypi:~ $ sudo mount -t auto /dev/sda2 /media/NASDrive
pi@raspberrypi:~ $ 
```

Enter the file and put NO in READ ONLY: **sudo nano /etc/samb.conf**



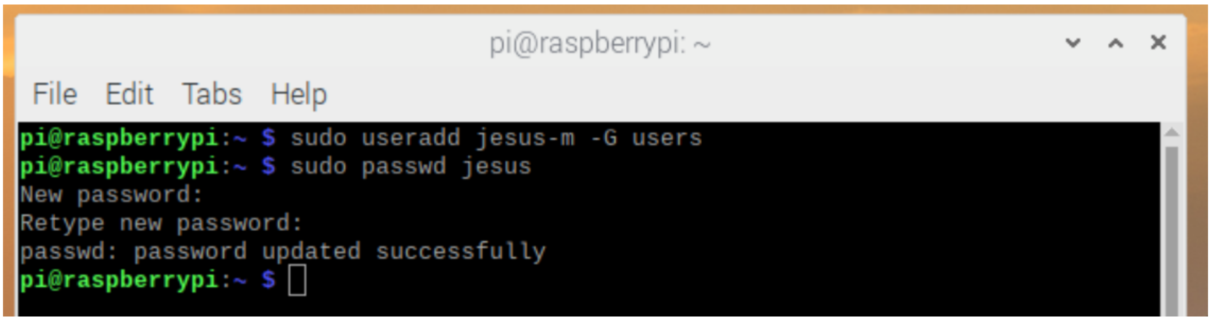
```
pi@raspberrypi:~ $ sudo nano /etc/samba/smb.conf
pi@raspberrypi:~ $ 
```

Type the command **sudo service smbd restart** to restart samba.



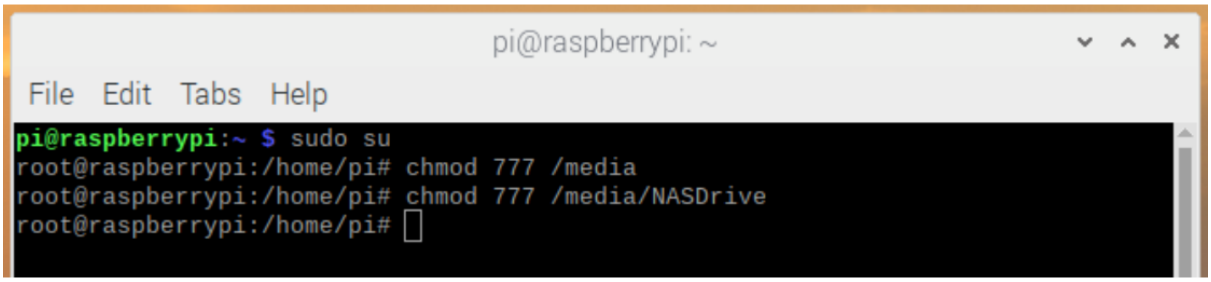
```
pi@raspberrypi:~ $ sudo service smbd restart
pi@raspberrypi:~ $ 
```

Create a password and username and add it to samba: **sudo useradd jesus-m -G users**



```
pi@raspberrypi:~ $ sudo useradd jesus-m -G users
pi@raspberrypi:~ $ sudo passwd jesus
New password:
Retype new password:
passwd: password updated successfully
pi@raspberrypi:~ $ 
```

The permissions need to be configured to allow every device to access the files, however, since the username was created and added to samba in the previous step, only that username will be allowed to see the shared folder. Type: **sudo chmod 777 /media/NASDrive**



```
pi@raspberrypi:~ $ sudo su
root@raspberrypi:/home/pi# chmod 777 /media
root@raspberrypi:/home/pi# chmod 777 /media/NASDrive
root@raspberrypi:/home/pi# 
```

Challenges and milestones

The group have encountered any challenges along the way to finally install the VPN server on the Raspberry Pi and connect to it as a client. Those can be classified in some categories such as Infrastructure and Resources, Network, Corona virus and Healthy issues.

Infrastructure and Resources

The initial challenge faced by the group smart5 is related to the basic resources that were required to accomplish the task, for example, monitor and USB keyboard. All of the components of the group are International students which rent rooms and share accommodation with other people. For this reason, the group had to ask the supervisor for a monitor that we could use for the initial configuration of the raspberry Pi. If the installation could not be done in one day, the group would have to decide who would carry on with the task and how to get a monitor to do it. Another problem is the fact that no one has a car in the group, so bringing the monitor around was not a practical solution.

The problem was partially solved by the use of Putty which allows the user to access the Raspberry Pi remotely by using Secure Shell. However, each time the Raspberry Pi device would be connected to a different network the monitor was needed again for the IP address configuration. In order to use Putty, it is necessary to assign a static IP address to the device, therefore, every time the network was different the group faced the issue with keyboard and keyboard.

Another milestone that the group came across was regards to the steps for the installation process. The idea of installing a VPN server on a Raspberry Pi looks simple at prior, but the reality is different. There are many tutorials and websites guiding you through the process, however, most of them are obsolete and do not work in the newest version of Raspbian. The group tried different tutorials, but it failed many times due to the commands that were not updates or files that do not exist anymore.

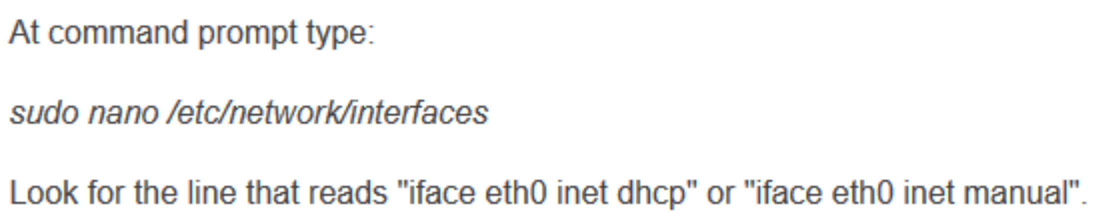
The biggest problem with the old tutorial is that the person trying to install the server only finds out about the issue when it is in the middle or at the end. After typed a bunch of commands and get nearly to the end of the tutorial the user might find out that the folder does not exist anymore or it simply do not work. Troubleshooting and going on IT experts pages and Linux forums to find the solution was done countless times by all of the components of group, but another problem was created by trying to mix up different suggestions about how to solve the issue, because after it, it was even more difficult to troubleshoot the issues.

Furthermore, the newest tutorials about the installation are the ones which use a script based on the PiVPN protocol. Using the script is so much handier and easy, but the idea behind this project is going through each step of the process in order to understand it completely.

Moreover, the students were not very familiar with the Linux operating system and had to learn some of commands used in Linux and understand how to navigate among files and folders. For this reason, troubleshooting was also considered a challenge that was accomplished at the end.

The first time the implementation was tried, the tutorial *How to set up your own Raspberry Pi powered VPN* by Kate Russel, posted on the BBC website, was chosen as the installation guide, because of the full description of commands and the clarity of the process, however, along the steps it became clear that a new tutorial would have to be used due to the fact that many commands were already out of date since that article was written in 2015.

The problem started at the beginning, with the command to set the IP address to be STATIC instead of dynamic. The tutorial instructed:

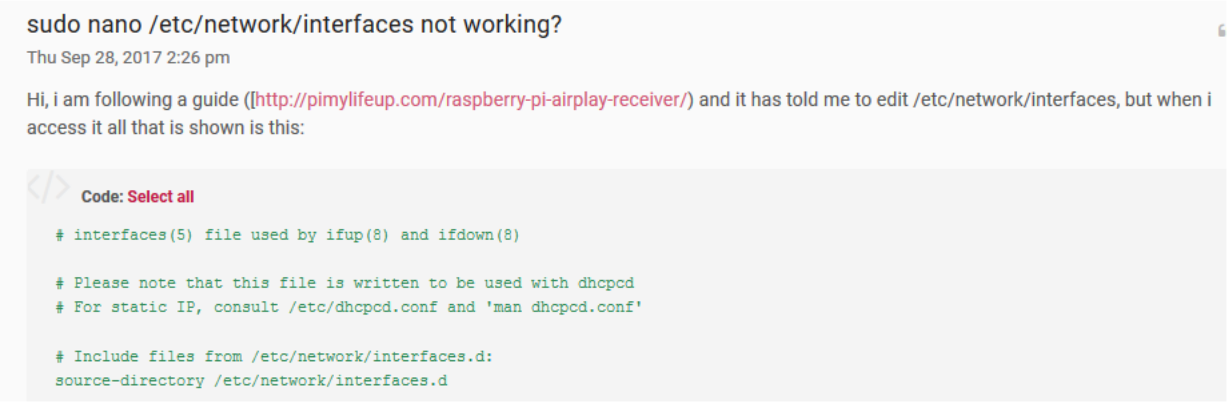


At command prompt type:

sudo nano /etc/network/interfaces

Look for the line that reads "iface eth0 inet dhcp" or "iface eth0 inet manual".

Once we typed the command, we found out that this file did not existed due to an update in the Raspbian operating system. For those, reason we would have to find an alternative way to configure it.



sudo nano /etc/network/interfaces not working?

Thu Sep 28, 2017 2:26 pm

Hi, i am following a guide ([http://pimylifeup.com/raspberry-pi-airplay-receiver/) and it has told me to edit /etc/network/interfaces, but when i access it all that is shown is this:

Code: Select all

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
```

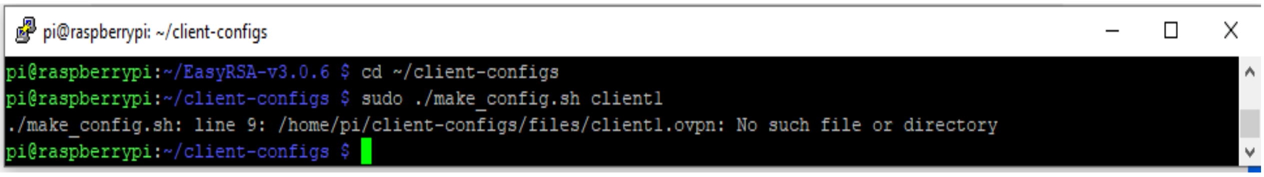
In order the issue, the group went to forums and websites to find a command to replace the previous one. The video *Assign a static IP Address to the Raspberry Pi* posted by ErrorAvoid channel on YouTube website was used and the command was used instead.

Type **sudo nano /etc/dhcpcd.conf** and add the following lines at the top of the file (as it was shown with more details in Chapter 4).

```
Interface wan0(eth0)
Static ip_address= [ip address]/[subnet]
Static routers= [ip address]
Static domain_name_servers= [IP address of the DNS
```

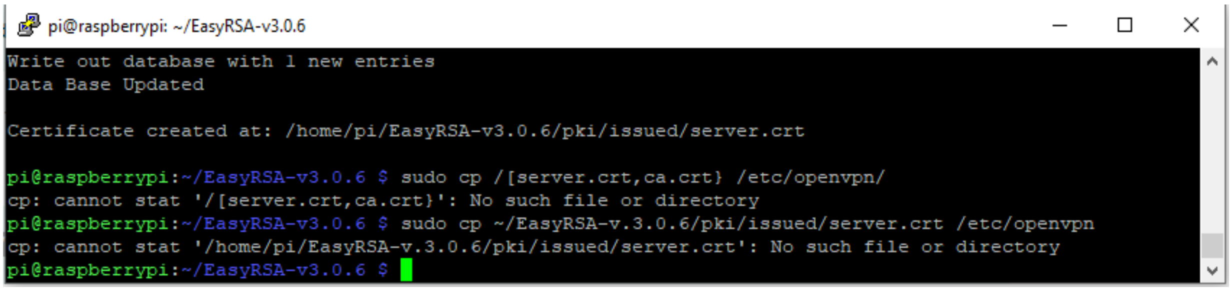
We fixed the issue and could continue, but during the whole process we came across different errors messages that we had to troubleshoot and find a solution. Some screenshots below illustrate some of the errors:

Example 1



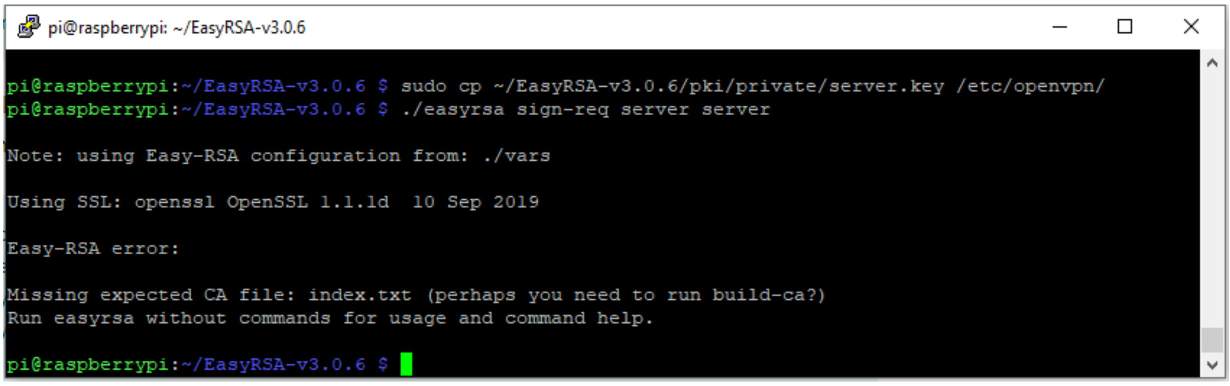
```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ cd ~/client-configs
pi@raspberrypi:~/client-configs $ sudo ./make_config.sh client1
./make_config.sh: line 9: /home/pi/client-configs/files/client1.ovpn: No such file or directory
pi@raspberrypi:~/client-configs $
```

Example 2



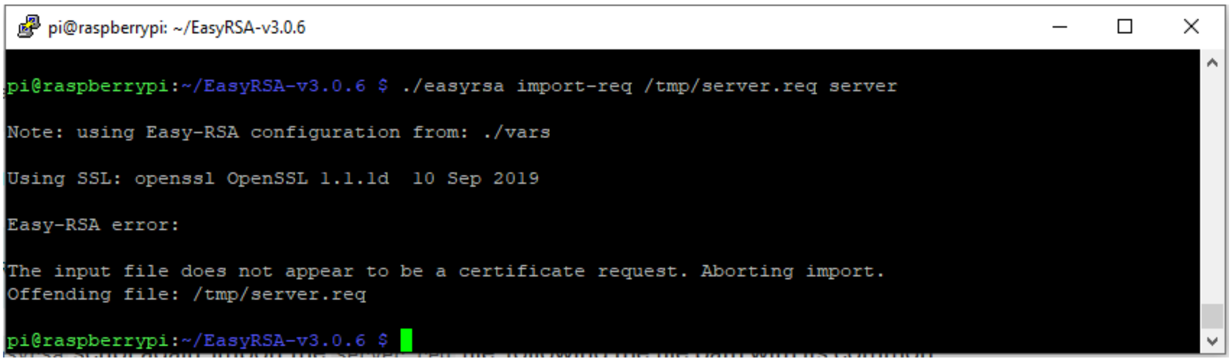
```
Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/pi/EasyRSA-v3.0.6/pki/issued/server.crt

pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo cp /[server.crt,ca.crt} /etc/openvpn/
cp: cannot stat '/[server.crt,ca.crt}': No such file or directory
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo cp ~/EasyRSA-v.3.0.6/pki/issued/server.crt /etc/openvpn
cp: cannot stat '/home/pi/EasyRSA-v.3.0.6/pki/issued/server.crt': No such file or directory
pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

Example 3



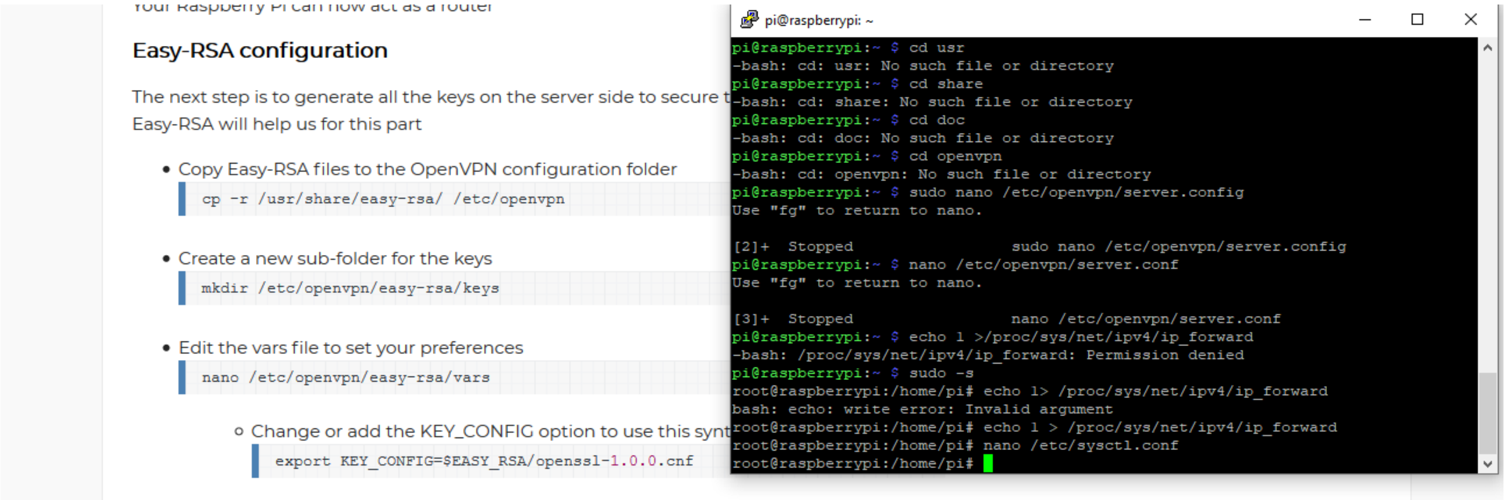
```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ sudo cp ~/EasyRSA-v3.0.6/pki/private/server.key /etc/openvpn/
pi@raspberrypi:~/EasyRSA-v3.0.6 $ ./easyrsa sign-req server server

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019

Easy-RSA error:

Missing expected CA file: index.txt (perhaps you need to run build-ca?)
Run easyrsa without commands for usage and command help.

pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

Example 4


pi@raspberrypi: ~/EasyRSA-v3.0.6

```
pi@raspberrypi:~/EasyRSA-v3.0.6 $ ./easyrsa import-req /tmp/server.req server

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019

Easy-RSA error:

The input file does not appear to be a certificate request. Aborting import.
Offending file: /tmp/server.req

pi@raspberrypi:~/EasyRSA-v3.0.6 $
```

Example 5


Your Raspberry Pi can now act as a router

Easy-RSA configuration

The next step is to generate all the keys on the server side to secure t
Easy-RSA will help us for this part

- Copy Easy-RSA files to the OpenVPN configuration folder

```
cp -r /usr/share/easy-rsa/ /etc/openvpn
```

- Create a new sub-folder for the keys

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

- Edit the vars file to set your preferences

```
nano /etc/openvpn/easy-rsa/vars
```

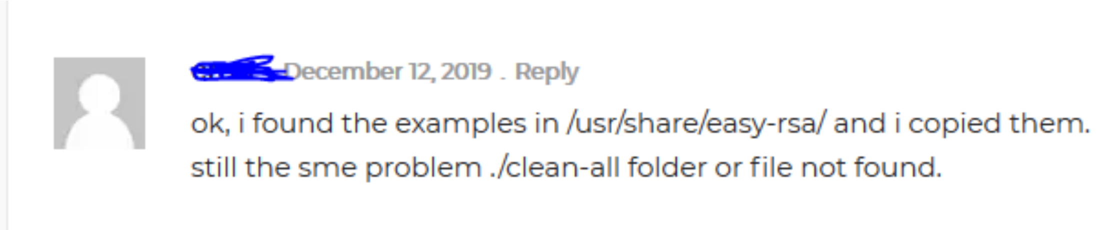
  - Change or add the KEY_CONFIG option to use this synt

```
export KEY_CONFIG=$EASY_RSA/openssl-1.0.0.cnf
```

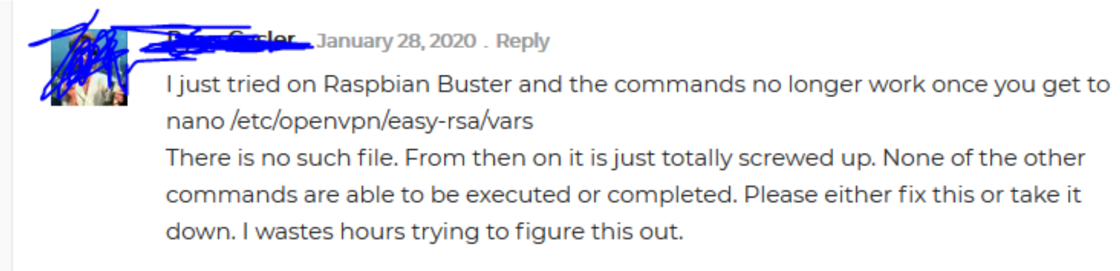
pi@raspberrypi: ~

```
pi@raspberrypi:~ $ cd usr
-bash: cd: usr: No such file or directory
pi@raspberrypi:~ $ cd share
-bash: cd: share: No such file or directory
pi@raspberrypi:~ $ cd doc
-bash: cd: doc: No such file or directory
pi@raspberrypi:~ $ cd openvpn
-bash: cd: openvpn: No such file or directory
pi@raspberrypi:~ $ sudo nano /etc/openvpn/server.config
Use "fg" to return to nano.

[2]+  Stopped                 sudo nano /etc/openvpn/server.config
pi@raspberrypi:~ $ nano /etc/openvpn/server.conf
Use "fg" to return to nano.

[3]+  Stopped                 nano /etc/openvpn/server.conf
pi@raspberrypi:~ $ echo 1 >/proc/sys/net/ipv4/ip_forward
-bash: /proc/sys/net/ipv4/ip_forward: Permission denied
pi@raspberrypi:~ $ sudo -s
root@raspberrypi:/home/pi# echo 1> /proc/sys/net/ipv4/ip_forward
bash: echo: write error: Invalid argument
root@raspberrypi:/home/pi# echo 1 > /proc/sys/net/ipv4/ip_forward
root@raspberrypi:/home/pi# nano /etc/sysctl.conf
root@raspberrypi:/home/pi#
```

Other users were having the same issues as the smart5 group. They posted those comments in the forum: https://raspberrytips.com/install-openvpn-raspberry-pi/


[redacted] December 12, 2019 . Reply

ok, i found the examples in /usr/share/easy-rsa/ and i copied them.
still the sme problem ./clean-all folder or file not found.


[redacted] January 28, 2020 . Reply

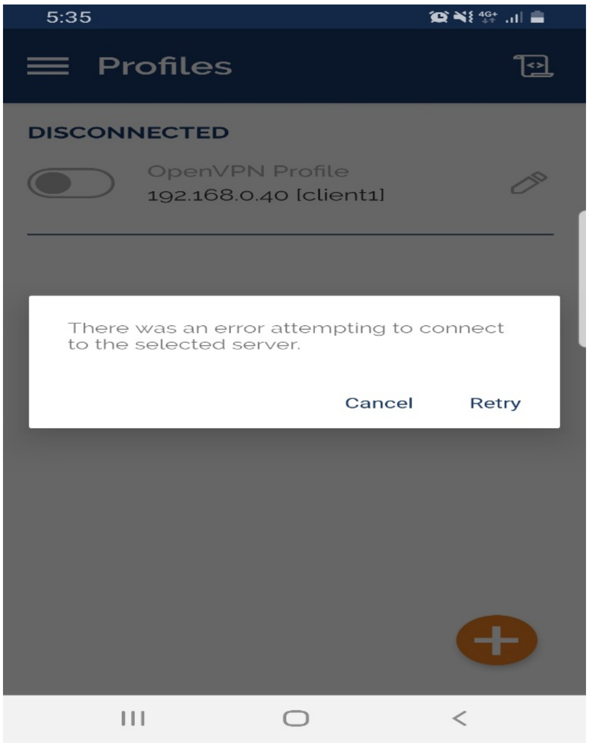
I just tried on Raspbian Buster and the commands no longer work once you get to nano /etc/openvpn/easy-rsa/vars
There is no such file. From then on it is just totally screwed up. None of the other commands are able to be executed or completed. Please either fix this or take it down. I wastes hours trying to figure this out.

It was really important to troubleshoot and find solutions for the out of date commands or the other issues encountered during the process, however, mixing up tutorials with solutions from other guides created a bigger problem to troubleshoot the other next problems that occurred. It was hard to tell where the error was. It was like following three recipes to make a cake and at the end having to find out how to fix the tasteless mixture without knowing which ingredient was missing or was in excess.

Facing so many troubles during the process, helped the team to understand the process better and chose guides more wisely. By looking at the commands of the guide, it was possible to tell if it was worth to give a try or if it was better to keep searching for more information.

This process was important and part of the challenge, however, it slowed down the implementation plan, because in the original schedule the installation would not take more than 4 weeks and actually it needed nearly the double of the amount of time initially planned.

Even after setting up the OpenVPN there were problems to connect to it that it added up a bit more of challenges to the task. On the image below it is possible to see that the connection could not be done. After that, the group realised that the IP address being used was the private IP address instead of the public IP address. It was a simple mistake, that once fixed help the members to better understand the project.



DISCONNECTED

OpenVPN Profile
192.168.0.40 [client1]

There was an error attempting to connect to the selected server.

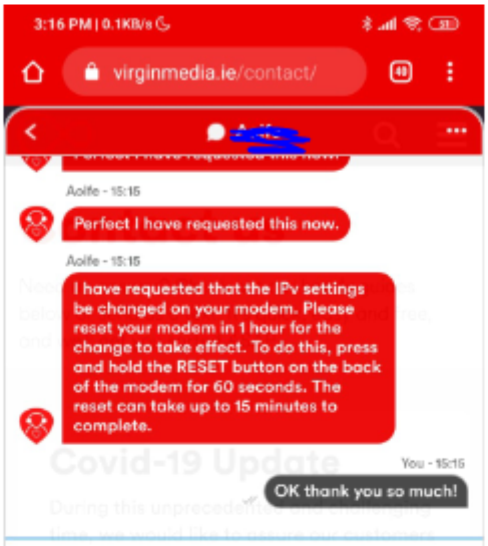
Network

The network was considered a challenge due to the online meetings that replaced the face to face interaction during the lockdown. Some members of the group had problems with their connection and had problems during the meeting and ended missing some information that were discussed with the supervisor.

Other issue related to the network was regards to the configuration for the VPN. In chapter 4, it was mentioned that port forwarding should be enable in the home router to allow the network traffic go through the port 1194 to the Raspberry Pi. However, some ISP, such as, Virgin Media, blocks this functionality for security reasons.

In order to enable this setting, it was necessary to contact Virgin Media to make the request, however, the request could only be processed after the billing person permission and this person was none of the members of the group.

After some conversations, one of members convinced his roommate to make the request with Virgin Media, so the project could be continued. The image bellow shows the conversation between them about the request to Virgin Media.



[redacted]

Aoife - 15:15

Perfect I have requested this now.

Aoife - 15:15

I have requested that the IPv settings be changed on your modem. Please reset your modem in 1 hour for the change to take effect. To do this, press and hold the RESET button on the back of the modem for 60 seconds. The reset can take up to 15 minutes to complete.

You - 15:15

OK thank you so much!

Image to illustrate the conversation about Virgin Media request.

Corona Virus and Health issues

Another challenge that it is worth to mention are the Health issues that two members of the group had during the semester. One of them had to travel to its home country for a surgery and stayed away for three weeks. The other member had an accident while on holidays and had to stay in the hospital for a few weeks.

Although some arrangements were made to address those situations, some tasks were delayed for a couple of weeks from the initial plan.

The corona virus represents also a challenge for the members of the groups. Not only for the group smart5, but all the other students, teachers and people around the world who felt the impact of a world pandemic in their professional and personal life.

The social distancing measure imposed by the Government made impossible to have personal meetings where the members could discuss some ideas or issues and go through the project together.

Skype and Google Hangouts were very important for this period because they allowed the group to continue with the meeting and with the sessions with the supervisor. However, sometimes the poor connectivity affected the online sessions for some of the members.

Other challenge brought by the Corona Virus was the fact that many places were closed down, including the library at CCT Dublin College. Since the beginning of the project, many books were used as a resource for the research and not having it for the second part was a bit of challenge. There are some many contents available on the web nowadays, but the traditional was of looking for information in books always aggregates more knowledge and credibility to the projects.

Chapter 5 – Testing and Evaluation

This chapter shows the results of the implementation from the previous chapter. It uses Wireshark and other resources to test the effectiveness of the process. Screenshots are used to document the results along with some comments explaining them.

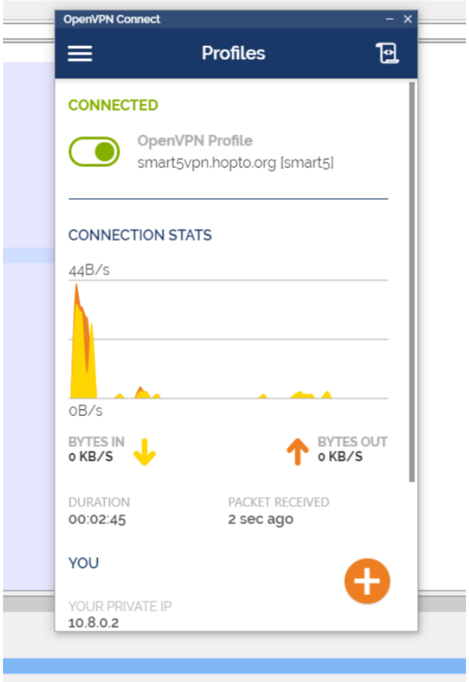
Connecting the client to OpenVPN server

The first step of testing is to connect the client to the OpenVPN server. Fort that, it is necessary to download the OpenVPN client to the device that will be connected to the VPN server.

Go to the web page: https://openvpn.net/client-connect-vpn-for-windows/ to download OpenVPN for Windows machines. If the users want to connect to the VPN using the mobile phone, it can download OpenVPN client in the Google Store or Apple Store for MAC users. The app is called OpenVPN client connect.

Once the download is ready, the user has to transfer the **client.ovpn** file to their device that will be connected to the VPN. For that, the user can use a software called WinSCP to transfer files from the Raspberry Pi to the device using a secure protocol. The SCP extends for Secure Copy Protocol and it helps to transfer files securely from a local to a remote host using the port 22. It is similar to File Transfer Protocol (FTP), but it includes security and authentication to the process though the use of the SSH protocol (Wilson, 2019). An alternative way is just to copy the **client.ovpn** file to an USB stick from the Raspberry Pi and transfer it to the laptop.

Once the file is transferred, run the OpenVPN client program and import the file with the certificates and keys to authenticate the user. Click connect. When the user connects to the VPN, the connection status turns green in the Graphic user Interface of the OpenVPN program.



CONNECTED

OpenVPN Profile
smart5vpn.hopto.org [smart5]

CONNECTION STATS

BYTES IN 0 KB/S

BYTES OUT 0 KB/S

DURATION 00:02:45

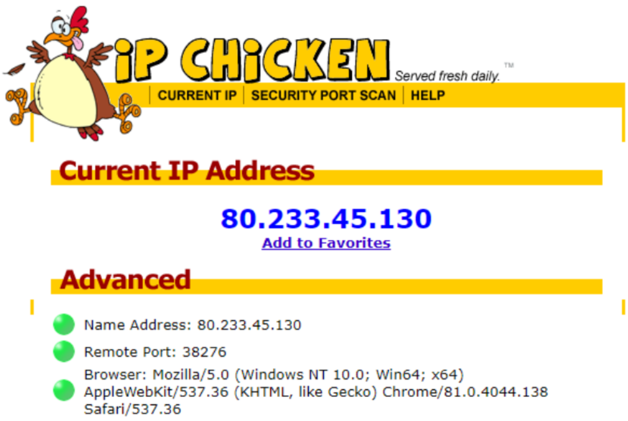
PACKET RECEIVED 2 sec ago

YOU

YOUR PRIVATE IP 10.8.0.2

The OpenVPN server is on the network 10.8.0.0 and has the default IP address of 10.8.0.1. It dynamically assigns IP addresses to the devices that are connected to it, according to the configuration that was done in the configuration file of the server in Chapter 4.

In order to verify that the network traffic is going through the VPN server, the user can go to a website to check if the IP address has changed. For example, before connecting to the VPN server, the public IP address of the device provided by the ISP was 80.233.45.130 while the private IP address was 172.20.10.3. This can be verified on the website IPCHICKEN.

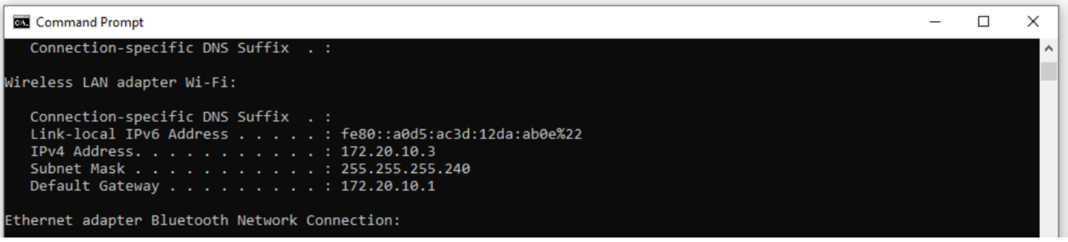

IP CHiCKEN Served fresh daily. ™
CURRENT IP | SECURITY PORT SCAN | HELP

Current IP Address

80.233.45.130
Add to Favorites

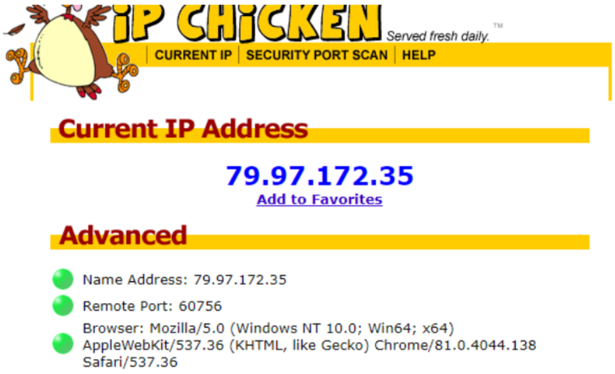
Advanced

- Name Address: 80.233.45.130
- Remote Port: 38276
- Browser: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36



```
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::a0d5:ac3d:12da:ab0e%22
   IPv4 Address. . . . . . . . . . . : 172.20.10.3
   Subnet Mask . . . . . . . . . . . : 255.255.255.240
   Default Gateway . . . . . . . . . : 172.20.10.1

Ethernet adapter Bluetooth Network Connection:
```

Once the device is connected to the VPN server, the public IP address you should see is now the one from your home network. The reason for that, it is due to the tunnel that it is created between the two devices provided by the OpenVPN protocol. The network traffic goes through the tunnel, to the VPN server and then goes out to the internet through your Home Network that has different public IP from the clients network. Once again, the verification can be done by using the website IPChicken. The public IP address is now 79.97.172.35 and the IP address which the OpenVPN dynamically assigned to the device is 10.8.0.2.

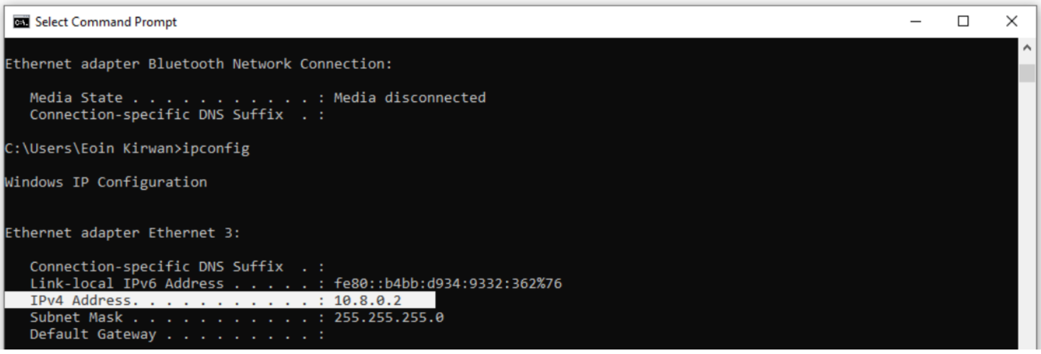


Current IP Address

79.97.172.35
Add to Favorites

Advanced

● Name Address: 79.97.172.35
● Remote Port: 60756
● Browser: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36

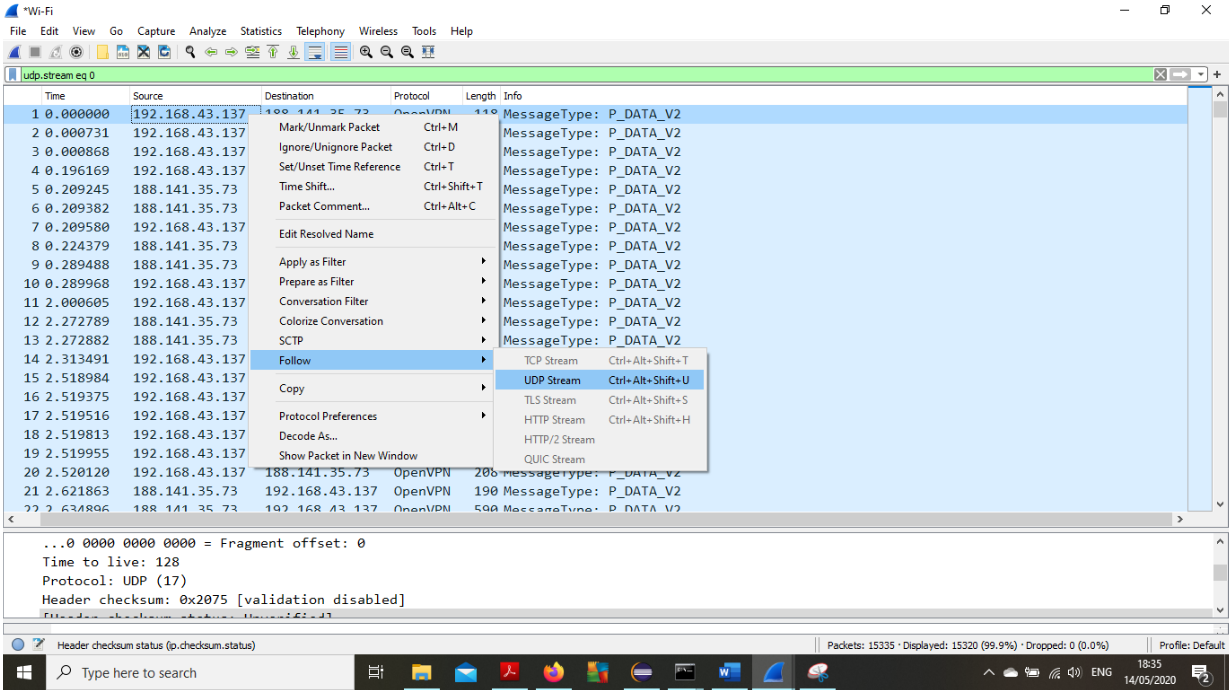


```
Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\Eoin Kirwan>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet 3:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::b4bb:d934:9332:362%76
   IPv4 Address. . . . . . . . . . . : 10.8.0.2
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

Testing VPN Connectivity by using Wireshark

One of the best ways to test the use of VPN is using a packet analyser software, such as Wireshark, in order to see the network traffic. One of the features that can be analysed is if VPN is encrypting the data.

In order to verify that, the packet number 1 with source IP address 192.168.43.137 was chosen. The option follow > UDP stream was selected in order to see the contents of the packets.


*Wi-Fi
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream
Type here to search
ENG
18:35
14/05/2020

```
udp.stream eq 0
```

```
     Time      Source          Destination     Protocol  Length Info
   1 0.000000  192.168.43.137  188.141.35.73   OpenVPN   118 MessageType: P_DATA_V2
   2 0.000731  192.168.43.137                                 MessageType: P_DATA_V2
   3 0.000868  192.168.43.137                                 MessageType: P_DATA_V2
   4 0.196169  192.168.43.137                                 MessageType: P_DATA_V2
   5 0.209245  188.141.35.73                                  MessageType: P_DATA_V2
   6 0.209382  188.141.35.73                                  MessageType: P_DATA_V2
   7 0.209580  192.168.43.137                                 MessageType: P_DATA_V2
   8 0.224379  188.141.35.73                                  MessageType: P_DATA_V2
   9 0.289488  188.141.35.73                                  MessageType: P_DATA_V2
  10 0.289968  192.168.43.137                                 MessageType: P_DATA_V2
  11 2.000605  192.168.43.137                                 MessageType: P_DATA_V2
  12 2.272789  188.141.35.73                                  MessageType: P_DATA_V2
  13 2.272882  188.141.35.73                                  MessageType: P_DATA_V2
  14 2.313491  192.168.43.137
  15 2.518984  192.168.43.137
  16 2.519375  192.168.43.137
  17 2.519516  192.168.43.137
  18 2.519813  192.168.43.137
  19 2.519955  192.168.43.137
  20 2.520120  192.168.43.137  188.141.35.73   OpenVPN   208 MessageType: P_DATA_V2
  21 2.621863  188.141.35.73   192.168.43.137  OpenVPN   190 MessageType: P_DATA_V2
  22 2.634896  188.141.35.73   192.168.43.137  OpenVPN   590 MessageType: P_DATA_V2
```

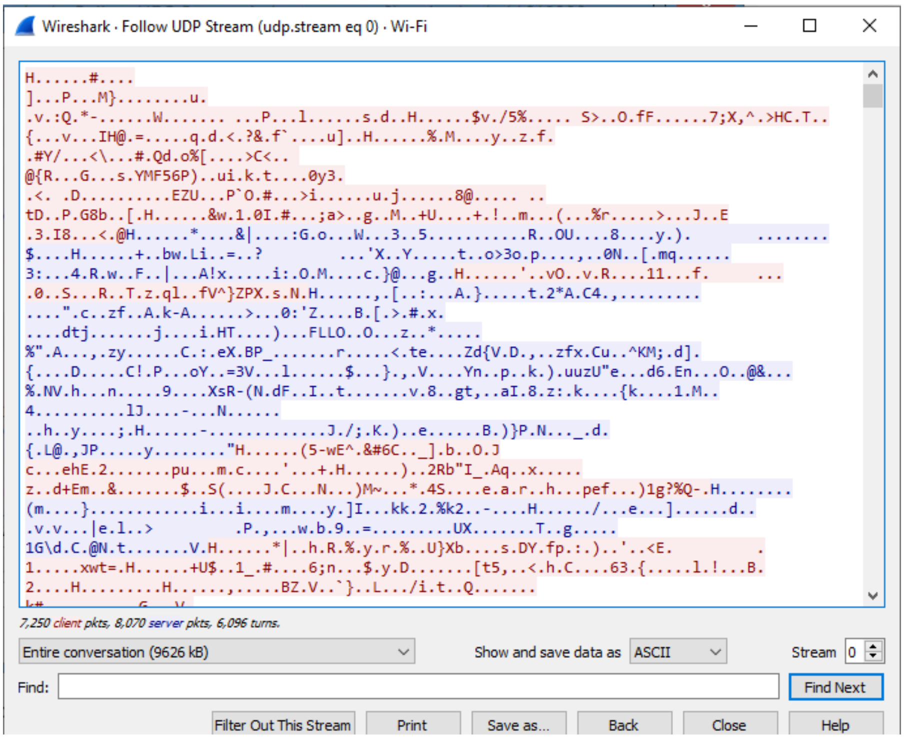
```
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x2075 [validation disabled]
    [Header checksum status: Unverified]
```

Header checksum status (ip.checksum.status)

Packets: 15335 · Displayed: 15320 (99.9%) · Dropped: 0 (0.0%)

Profile: Default

The data inside the full stream was unreadable which confirms that the VPN is encrypting the data. As discussed in Chapter 1 and in the implementation in Chapter 4, OpenVPN provides cryptographic algorithms that encrypt the data inside the tunnel.



```
H......#....
]...P...M}........u.
.v.:Q.*-......W....... ...P...l......s.d..H......$v./5%..... S>..O.fF......7;X,^.>HC.T..
{...v...IH@.=.....q.d.<.?&.f`....u]..H......%.M....y..z.f.
.#Y/...<\...#.Qd.o%[....>C<..
@{R...G...s.YMF56P)..ui.k.t....0y3.
.<. .D..........EZU...P`O.#...>i......u.j......8@..... ..
tD..P.G8b..[.H......&w.1.0I.#...;a>..g..M..+U....+.!..m...(...%r.....>...J..E
.3.I8...<.@H......*....&|....:G.o...W...3..5...........R..OU....8....y.).       ........
$....H......+..bw.Li..=..?          ...'X..Y.....t..o>3o.p....,..0N..[.mq......
3:...4.R.w..F..|...A!x.....i:.O.M....c.}@...g..H......'..vO..v.R....11...f.      ...
.0..S...R..T.z.ql..fV^}ZPX.s.N.H......,.[..:...A.}.....t.2*A.C4.,.........
....".c..zf..A.k-A......>...0:'Z....B.[.>.#.x.
....dtj.......j....i.HT....)...FLLO..O...z..*.....
%".A...,.zy......C.:.eX.BP_.......r.....<.te....Zd{V.D.,..zfx.Cu..^KM;.d].
{....D.....C!.P...oY..=3V...l......$...}.,.V....Yn..p..k.).uuzU"e...d6.En...O..@&...
%.NV.h...n.....9....XsR-(N.dF..I..t.......v.8..gt,..aI.8.z:.k....{k....1.M..
4..........lJ....-...N......
..h..y....;.H......-.............J./;.K.)..e......B.)}P.N..._.d.
{.L@.,JP.....y........"H......(5-wE^.&#6C.._].b..O.J
c...ehE.2.......pu...m.c....'...+.H......)..2Rb"I_.Aq..x.....
z..d+Em..&.......$..S(....J.C...N...)M~...*.4S....e.a.r..h...pef...)1g?%Q-.H........
(m....}............i...i....m....y.]I...kk.2.%k2..-....H....../...e...]......d..
.v.v...|e.l..>          .P.,...w.b.9..=.........UX.......T..g.....
1G\d.C.@N.t.......V.H......*|..h.R.%.y.r.%..U}Xb....s.DY.fp.:.)..'..<E.         .
1.....xwt=.H......+U$..1_.#....6;n...$.y.D.......[t5,..<.h.C....63.{.....l.!...B.
2....H.........H......,.....BZ.V..`}..L.../i.t..Q.......
```

7,250 client pkts, 8,070 server pkts, 6,096 turns.

Entire conversation (9626 kB)

Show and save data as: ASCII

Stream: 0

Find:

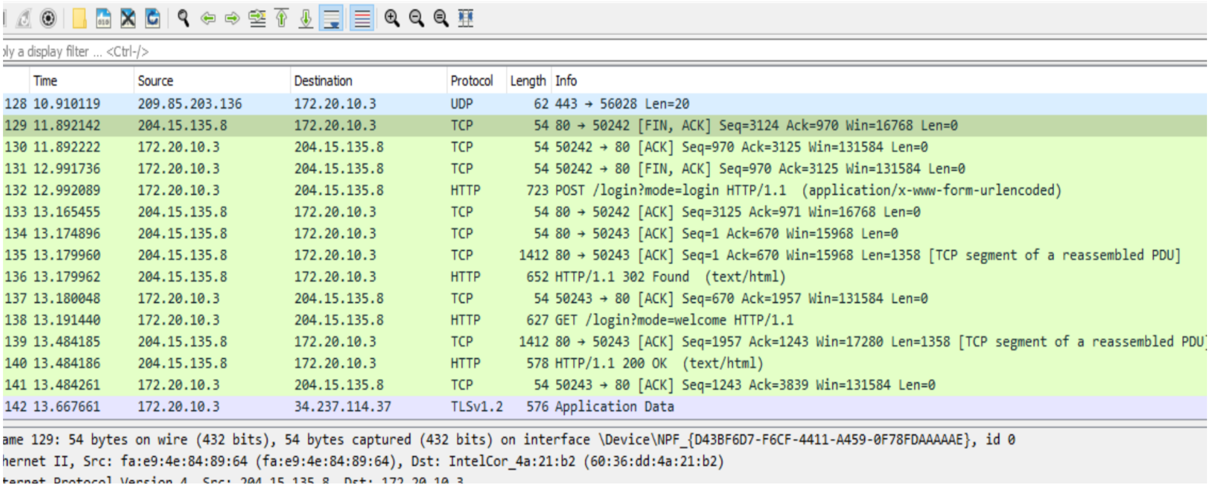
Another way of testing the VPN is by analysing the data that is sent to a website which uses HTTP instead of HTTPS. HTTP stands for Hypertext Transfer Protocol and it [definir]. Meanwhile, HTTPS uses Secure Shell for encryption. For this reason, the data that is sent using HTTPS is secure, while the data HTTP is not secure.

In order to analyse the usage and advantages of using a VPN, the website **testing-ground.scraping.pro/login** is going to be used to see the data for username and password which will be sent to their server without the encryption.

On the screenshot bellow, it is possible to see that the Transmission Control Protocol (TCP) is being used to establish a connection with the website **testing-ground.scraping.pro/login**(204.15.135.8). The TCP protocol is responsible to define the rules for the communication of two devices across the network on a process known as 3-way handshake (InetDaemon, 2018). The process can be described:

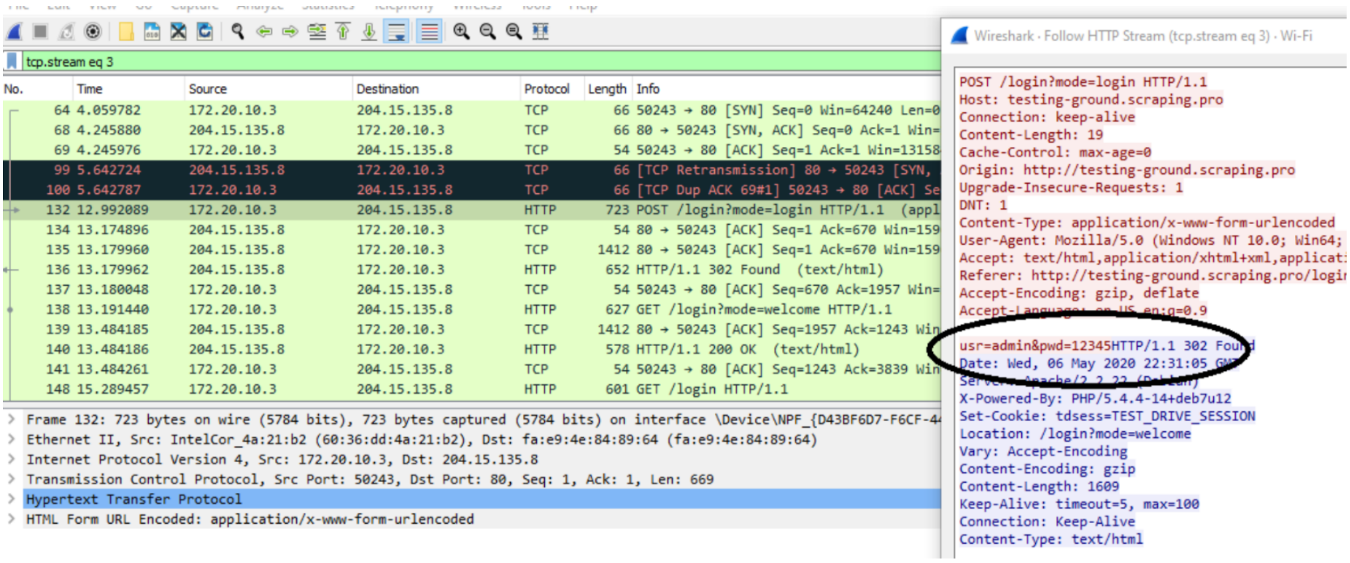
- The device 172.20.10.3 sends a TCP SYN packet to 204.15.135.8 to say “hello”.
- 204.15.135.8 sends a SYN-ACK packet to 172.20.10.3 to say: “yes, I am here”
- 172.20.10.3 sends an ACK packet back to 204.15.135.8 to say: “I will transfer now”
- Once 204.15.135.8 receives the ACK, they start to the communicate and exchange information.

The importance of this process is to guarantee that the packets will be transmitted.



```
    Time        Source          Destination     Protocol  Length Info
128 10.910119   209.85.203.136  172.20.10.3     UDP         62 443 → 56028 Len=20
129 11.892142   204.15.135.8    172.20.10.3     TCP         54 80 → 50242 [FIN, ACK] Seq=3124 Ack=970 Win=16768 Len=0
130 11.892222   172.20.10.3     204.15.135.8    TCP         54 50242 → 80 [ACK] Seq=970 Ack=3125 Win=131584 Len=0
131 12.991736   172.20.10.3     204.15.135.8    TCP         54 50242 → 80 [FIN, ACK] Seq=970 Ack=3125 Win=131584 Len=0
132 12.992089   172.20.10.3     204.15.135.8    HTTP       723 POST /login?mode=login HTTP/1.1  (application/x-www-form-urlencoded)
133 13.165455   204.15.135.8    172.20.10.3     TCP         54 80 → 50242 [ACK] Seq=3125 Ack=971 Win=16768 Len=0
134 13.174896   204.15.135.8    172.20.10.3     TCP         54 80 → 50243 [ACK] Seq=1 Ack=670 Win=15968 Len=0
135 13.179960   204.15.135.8    172.20.10.3     TCP       1412 80 → 50243 [ACK] Seq=1 Ack=670 Win=15968 Len=1358 [TCP segment of a reassembled PDU]
136 13.179962   204.15.135.8    172.20.10.3     HTTP       652 HTTP/1.1 302 Found  (text/html)
137 13.180048   172.20.10.3     204.15.135.8    TCP         54 50243 → 80 [ACK] Seq=670 Ack=1957 Win=131584 Len=0
138 13.191440   172.20.10.3     204.15.135.8    HTTP       627 GET /login?mode=welcome HTTP/1.1
139 13.484185   204.15.135.8    172.20.10.3     TCP       1412 80 → 50243 [ACK] Seq=1957 Ack=1243 Win=17280 Len=1358 [TCP segment of a reassembled PDU]
140 13.484186   204.15.135.8    172.20.10.3     HTTP       578 HTTP/1.1 200 OK  (text/html)
141 13.484261   172.20.10.3     204.15.135.8    TCP         54 50243 → 80 [ACK] Seq=1243 Ack=3839 Win=131584 Len=0
142 13.667661   172.20.10.3     34.237.114.37   TLSv1.2    576 Application Data

ame 129: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{D43BF6D7-F6CF-4411-A459-0F78FDAAAAAE}, id 0
hernet II, Src: fa:e9:4e:84:89:64 (fa:e9:4e:84:89:64), Dst: IntelCor_4a:21:b2 (60:36:dd:4a:21:b2)
ternet Protocol Version 4, Src: 204.15.135.8, Dst: 172.20.10.3
```

After the 3-handshake process, the communication starts and it is possible to see the use of the HTTP protocol. Once the packet is selected to be analysed, it is possible to see the information about the username **admin** and password **12345**. The reasons for that, as mentioned before, it is the fact that the website uses HTTP instead of HTTPS.

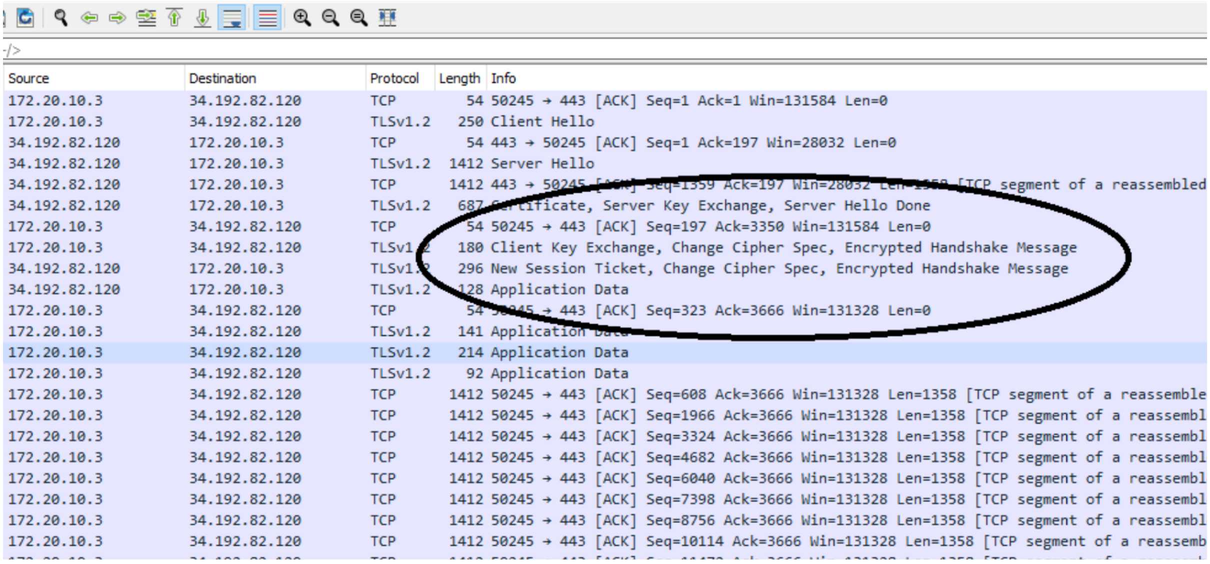


```
tcp.stream eq 3
No.  Time        Source          Destination     Protocol  Length Info
  64 4.059782    172.20.10.3     204.15.135.8    TCP         66 50243 → 80 [SYN] Seq=0 Win=64240 Len=0
  68 4.245880    204.15.135.8    172.20.10.3     TCP         66 80 → 50243 [SYN, ACK] Seq=0 Ack=1 Win=
  69 4.245976    172.20.10.3     204.15.135.8    TCP         54 50243 → 80 [ACK] Seq=1 Ack=1 Win=13158
  99 5.642724    204.15.135.8    172.20.10.3     TCP         66 [TCP Retransmission] 80 → 50243 [SYN,
 100 5.642787    172.20.10.3     204.15.135.8    TCP         66 [TCP Dup ACK 69#1] 50243 → 80 [ACK] Se
 132 12.992089   172.20.10.3     204.15.135.8    HTTP       723 POST /login?mode=login HTTP/1.1  (appl
 134 13.174896   204.15.135.8    172.20.10.3     TCP         54 80 → 50243 [ACK] Seq=1 Ack=670 Win=159
 135 13.179960   204.15.135.8    172.20.10.3     TCP       1412 80 → 50243 [ACK] Seq=1 Ack=670 Win=159
 136 13.179962   204.15.135.8    172.20.10.3     HTTP       652 HTTP/1.1 302 Found  (text/html)
 137 13.180048   172.20.10.3     204.15.135.8    TCP         54 50243 → 80 [ACK] Seq=670 Ack=1957 Win=
 138 13.191440   172.20.10.3     204.15.135.8    HTTP       627 GET /login?mode=welcome HTTP/1.1
 139 13.484185   204.15.135.8    172.20.10.3     TCP       1412 80 → 50243 [ACK] Seq=1957 Ack=1243 Win
 140 13.484186   204.15.135.8    172.20.10.3     HTTP       578 HTTP/1.1 200 OK  (text/html)
 141 13.484261   172.20.10.3     204.15.135.8    TCP         54 50243 → 80 [ACK] Seq=1243 Ack=3839 Win
 148 15.289457   172.20.10.3     204.15.135.8    HTTP       601 GET /login HTTP/1.1

> Frame 132: 723 bytes on wire (5784 bits), 723 bytes captured (5784 bits) on interface \Device\NPF_{D43BF6D7-F6CF-4
> Ethernet II, Src: IntelCor_4a:21:b2 (60:36:dd:4a:21:b2), Dst: fa:e9:4e:84:89:64 (fa:e9:4e:84:89:64)
> Internet Protocol Version 4, Src: 172.20.10.3, Dst: 204.15.135.8
> Transmission Control Protocol, Src Port: 50243, Dst Port: 80, Seq: 1, Ack: 1, Len: 669
> Hypertext Transfer Protocol
> HTML Form URL Encoded: application/x-www-form-urlencoded
```

Wireshark · Follow HTTP Stream (tcp.stream eq 3) · Wi-Fi

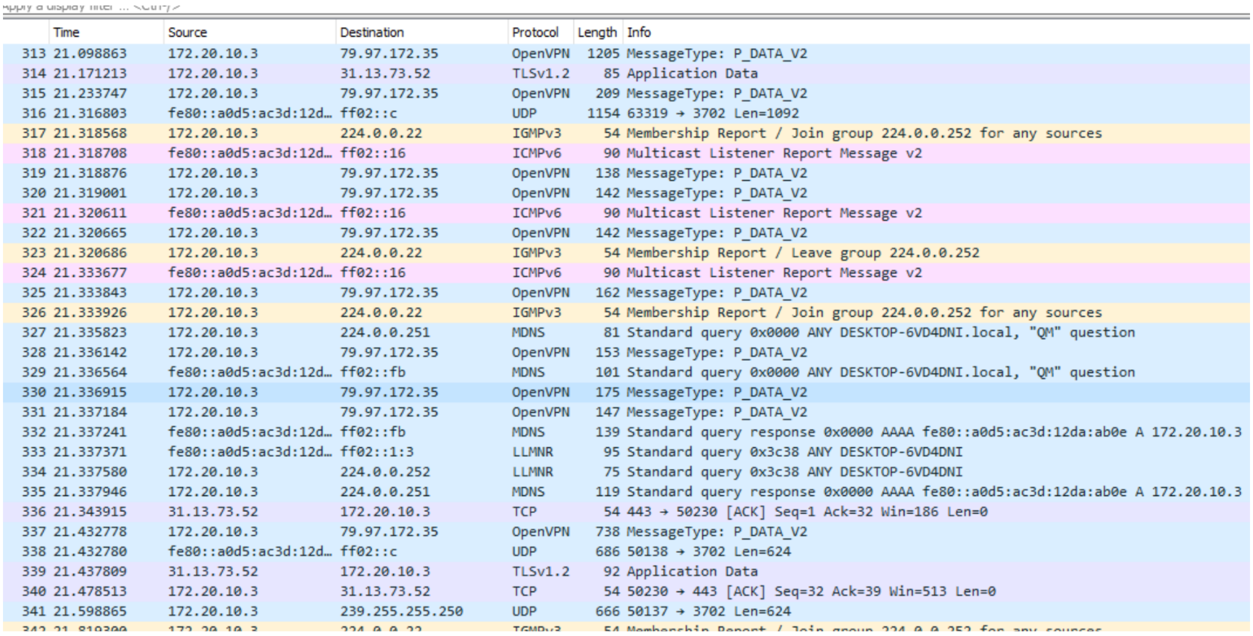
```
POST /login?mode=login HTTP/1.1
Host: testing-ground.scraping.pro
Connection: keep-alive
Content-Length: 19
Cache-Control: max-age=0
Origin: http://testing-ground.scraping.pro
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
Accept: text/html,application/xhtml+xml,applicat
Referer: http://testing-ground.scraping.pro/logi
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

usr=admin&pwd=12345HTTP/1.1 302 Found
Date: Wed, 06 May 2020 22:31:05 GMT
Server: Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.4.4-14+deb7u12
Set-Cookie: tdsess=TEST_DRIVE_SESSION
Location: /login?mode=welcome
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 1609
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

Now, the OpenVPN is being used and the data looks different from the previous screenshot. Firstly, it is possible to see that the TLS protocol is being used for the packets in the black circle. In chapter 4, it was mentioned that the TLS protocols is responsible to validate the client and the server by using the pre-shared key. The process below shows the authentication process on which the server and client exchange the pre-shared key.



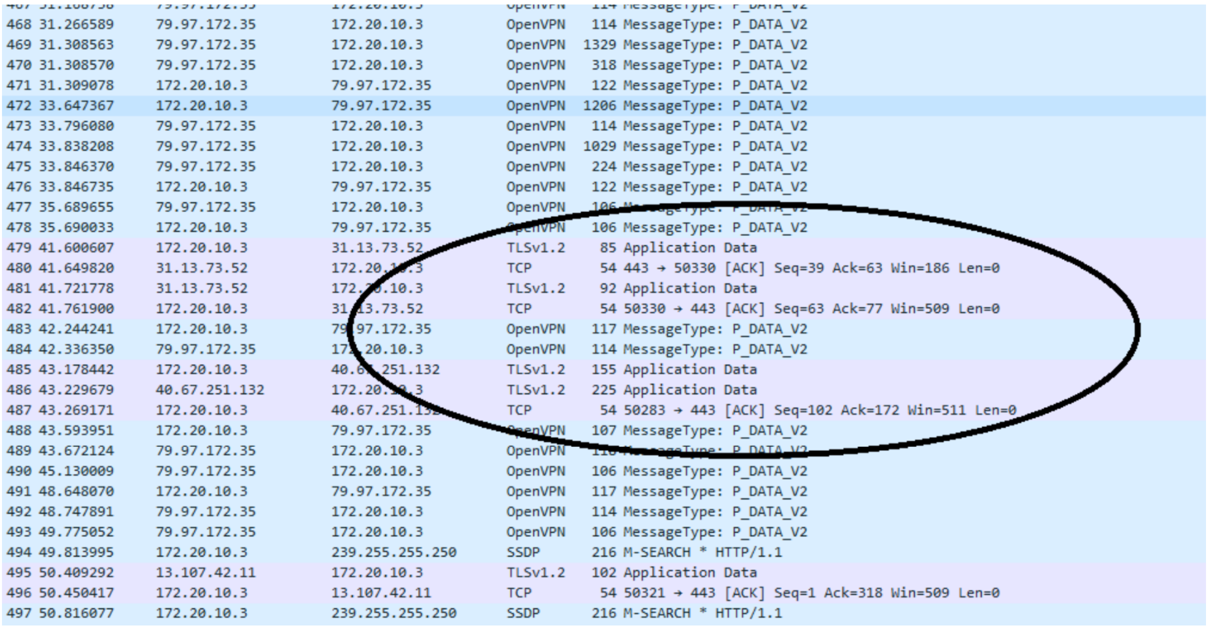
```
Source          Destination     Protocol  Length Info
172.20.10.3     34.192.82.120   TCP         54 50245 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=0
172.20.10.3     34.192.82.120   TLSv1.2    250 Client Hello
34.192.82.120   172.20.10.3     TCP         54 443 → 50245 [ACK] Seq=1 Ack=197 Win=28032 Len=0
34.192.82.120   172.20.10.3     TLSv1.2   1412 Server Hello
34.192.82.120   172.20.10.3     TCP       1412 443 → 50245 [ACK] Seq=1359 Ack=197 Win=28032 Len=1358 [TCP segment of a reassembled
34.192.82.120   172.20.10.3     TLSv1.2    687 Certificate, Server Key Exchange, Server Hello Done
172.20.10.3     34.192.82.120   TCP         54 50245 → 443 [ACK] Seq=197 Ack=3350 Win=131584 Len=0
172.20.10.3     34.192.82.120   TLSv1.2    180 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
34.192.82.120   172.20.10.3     TLSv1.2    296 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
34.192.82.120   172.20.10.3     TLSv1.2    128 Application Data
172.20.10.3     34.192.82.120   TCP         54 50245 → 443 [ACK] Seq=323 Ack=3666 Win=131328 Len=0
172.20.10.3     34.192.82.120   TLSv1.2    141 Application Data
172.20.10.3     34.192.82.120   TLSv1.2    214 Application Data
172.20.10.3     34.192.82.120   TLSv1.2     92 Application Data
172.20.10.3     34.192.82.120   TCP       1412 50245 → 443 [ACK] Seq=608 Ack=3666 Win=131328 Len=1358 [TCP segment of a reassemble
172.20.10.3     34.192.82.120   TCP       1412 50245 → 443 [ACK] Seq=1966 Ack=3666 Win=131328 Len=1358 [TCP segment of a reassembl
172.20.10.3     34.192.82.120   TCP       1412 50245 → 443 [ACK] Seq=3324 Ack=3666 Win=131328 Len=1358 [TCP segment of a reassembl
172.20.10.3     34.192.82.120   TCP       1412 50245 → 443 [ACK] Seq=4682 Ack=3666 Win=131328 Len=1358 [TCP segment of a reassembl
172.20.10.3     34.192.82.120   TCP       1412 50245 → 443 [ACK] Seq=6040 Ack=3666 Win=131328 Len=1358 [TCP segment of a reassembl
172.20.10.3     34.192.82.120   TCP       1412 50245 → 443 [ACK] Seq=7398 Ack=3666 Win=131328 Len=1358 [TCP segment of a reassembl
172.20.10.3     34.192.82.120   TCP       1412 50245 → 443 [ACK] Seq=8756 Ack=3666 Win=131328 Len=1358 [TCP segment of a reassembl
172.20.10.3     34.192.82.120   TCP       1412 50245 → 443 [ACK] Seq=10114 Ack=3666 Win=131328 Len=1358 [TCP segment of a reassemb
```

Here, the website **testing-ground.scraping.pro/login** is being accessed again. However, this time the VPN is being used. The credentials are the same as the previous section. It is not possible to see the HTTP protocol in the packets, instead it is possible to see the OpenVPN protocol being used.



```
Apply a display filter ... <Ctrl-/>
     Time         Source                  Destination          Protocol  Length Info
 313 21.098863    172.20.10.3             79.97.172.35         OpenVPN   1205 MessageType: P_DATA_V2
 314 21.171213    172.20.10.3             31.13.73.52          TLSv1.2     85 Application Data
 315 21.233747    172.20.10.3             79.97.172.35         OpenVPN    209 MessageType: P_DATA_V2
 316 21.316803    fe80::a0d5:ac3d:12d…    ff02::c              UDP       1154 63319 → 3702 Len=1092
 317 21.318568    172.20.10.3             224.0.0.22           IGMPv3      54 Membership Report / Join group 224.0.0.252 for any sources
 318 21.318708    fe80::a0d5:ac3d:12d…    ff02::16             ICMPv6      90 Multicast Listener Report Message v2
 319 21.318876    172.20.10.3             79.97.172.35         OpenVPN    138 MessageType: P_DATA_V2
 320 21.319001    172.20.10.3             79.97.172.35         OpenVPN    142 MessageType: P_DATA_V2
 321 21.320611    fe80::a0d5:ac3d:12d…    ff02::16             ICMPv6      90 Multicast Listener Report Message v2
 322 21.320665    172.20.10.3             79.97.172.35         OpenVPN    142 MessageType: P_DATA_V2
 323 21.320686    172.20.10.3             224.0.0.22           IGMPv3      54 Membership Report / Leave group 224.0.0.252
 324 21.333677    fe80::a0d5:ac3d:12d…    ff02::16             ICMPv6      90 Multicast Listener Report Message v2
 325 21.333843    172.20.10.3             79.97.172.35         OpenVPN    162 MessageType: P_DATA_V2
 326 21.333926    172.20.10.3             224.0.0.22           IGMPv3      54 Membership Report / Join group 224.0.0.252 for any sources
 327 21.335823    172.20.10.3             224.0.0.251          MDNS        81 Standard query 0x0000 ANY DESKTOP-6VD4DNI.local, "QM" question
 328 21.336142    172.20.10.3             79.97.172.35         OpenVPN    153 MessageType: P_DATA_V2
 329 21.336564    fe80::a0d5:ac3d:12d…    ff02::fb             MDNS       101 Standard query 0x0000 ANY DESKTOP-6VD4DNI.local, "QM" question
 330 21.336915    172.20.10.3             79.97.172.35         OpenVPN    175 MessageType: P_DATA_V2
 331 21.337184    172.20.10.3             79.97.172.35         OpenVPN    147 MessageType: P_DATA_V2
 332 21.337241    fe80::a0d5:ac3d:12d…    ff02::fb             MDNS       139 Standard query response 0x0000 AAAA fe80::a0d5:ac3d:12da:ab0e A 172.20.10.3
 333 21.337371    fe80::a0d5:ac3d:12d…    ff02::1:3            LLMNR       95 Standard query 0x3c38 ANY DESKTOP-6VD4DNI
 334 21.337580    172.20.10.3             224.0.0.252          LLMNR       75 Standard query 0x3c38 ANY DESKTOP-6VD4DNI
 335 21.337946    172.20.10.3             224.0.0.251          MDNS       119 Standard query response 0x0000 AAAA fe80::a0d5:ac3d:12da:ab0e A 172.20.10.3
 336 21.343915    31.13.73.52             172.20.10.3          TCP         54 443 → 50230 [ACK] Seq=1 Ack=32 Win=186 Len=0
 337 21.432778    172.20.10.3             79.97.172.35         OpenVPN    738 MessageType: P_DATA_V2
 338 21.432780    fe80::a0d5:ac3d:12d…    ff02::c              UDP        686 50138 → 3702 Len=624
 339 21.437809    31.13.73.52             172.20.10.3          TLSv1.2     92 Application Data
 340 21.478513    172.20.10.3             31.13.73.52          TCP         54 50230 → 443 [ACK] Seq=32 Ack=39 Win=513 Len=0
 341 21.598865    172.20.10.3             239.255.255.250      UDP        666 50137 → 3702 Len=624
 342 21.819300    172.20.10.3             224.0.0.22           IGMPv3      54 Membership Report / Join group 224.0.0.252 for any sources
```

It is possible to see again that the TLS protocol is being used to authenticate and initiate the session between the client and the VPN server.



```
 467 31.168758    79.97.172.35        172.20.10.3          OpenVPN    114 MessageType: P_DATA_V2
 468 31.266589    79.97.172.35        172.20.10.3          OpenVPN    114 MessageType: P_DATA_V2
 469 31.308563    79.97.172.35        172.20.10.3          OpenVPN   1329 MessageType: P_DATA_V2
 470 31.308570    79.97.172.35        172.20.10.3          OpenVPN    318 MessageType: P_DATA_V2
 471 31.309078    172.20.10.3         79.97.172.35         OpenVPN    122 MessageType: P_DATA_V2
 472 33.647367    172.20.10.3         79.97.172.35         OpenVPN   1206 MessageType: P_DATA_V2
 473 33.796080    79.97.172.35        172.20.10.3          OpenVPN    114 MessageType: P_DATA_V2
 474 33.838208    79.97.172.35        172.20.10.3          OpenVPN   1029 MessageType: P_DATA_V2
 475 33.846370    79.97.172.35        172.20.10.3          OpenVPN    224 MessageType: P_DATA_V2
 476 33.846735    172.20.10.3         79.97.172.35         OpenVPN    122 MessageType: P_DATA_V2
 477 35.689655    79.97.172.35        172.20.10.3          OpenVPN    106 MessageType: P_DATA_V2
 478 35.690033    172.20.10.3         79.97.172.35         OpenVPN    106 MessageType: P_DATA_V2
 479 41.600607    172.20.10.3         31.13.73.52          TLSv1.2     85 Application Data
 480 41.649820    31.13.73.52         172.20.10.3          TCP         54 443 → 50330 [ACK] Seq=39 Ack=63 Win=186 Len=0
 481 41.721778    31.13.73.52         172.20.10.3          TLSv1.2     92 Application Data
 482 41.761900    172.20.10.3         31.13.73.52          TCP         54 50330 → 443 [ACK] Seq=63 Ack=77 Win=509 Len=0
 483 42.244241    172.20.10.3         79.97.172.35         OpenVPN    117 MessageType: P_DATA_V2
 484 42.336350    79.97.172.35        172.20.10.3          OpenVPN    114 MessageType: P_DATA_V2
 485 43.178442    172.20.10.3         40.67.251.132        TLSv1.2    155 Application Data
 486 43.229679    40.67.251.132       172.20.10.3          TLSv1.2    225 Application Data
 487 43.269171    172.20.10.3         40.67.251.132        TCP         54 50283 → 443 [ACK] Seq=102 Ack=172 Win=511 Len=0
 488 43.593951    172.20.10.3         79.97.172.35         OpenVPN    107 MessageType: P_DATA_V2
 489 43.672124    79.97.172.35        172.20.10.3          OpenVPN    110 MessageType: P_DATA_V2
 490 45.130009    79.97.172.35        172.20.10.3          OpenVPN    106 MessageType: P_DATA_V2
 491 48.648070    172.20.10.3         79.97.172.35         OpenVPN    117 MessageType: P_DATA_V2
 492 48.747891    79.97.172.35        172.20.10.3          OpenVPN    114 MessageType: P_DATA_V2
 493 49.775052    79.97.172.35        172.20.10.3          OpenVPN    106 MessageType: P_DATA_V2
 494 49.813995    172.20.10.3         239.255.255.250      SSDP       216 M-SEARCH * HTTP/1.1
 495 50.409292    13.107.42.11        172.20.10.3          TLSv1.2    102 Application Data
 496 50.450417    172.20.10.3         13.107.42.11         TCP         54 50321 → 443 [ACK] Seq=1 Ack=318 Win=509 Len=0
 497 50.816077    172.20.10.3         239.255.255.250      SSDP       216 M-SEARCH * HTTP/1.1
```

It is important to mention that the usage of HTTPS and Secure Shell encrypts the data already without the participation of the VPN. So, if the previous website was using those protocols the information would not been readable, since it would be encrypted. The advantage of using the

VPN is the fact that it adds an extra layer of security and changes the headers of the packets. For this reason, an attacker trying to sniff the packets would not be able to see the protocols that are being used, the IP addresses, etc. The VPN not only provides encryption to the traffic, but it also protects the individual streams from the devices outside the tunnel (author, year).

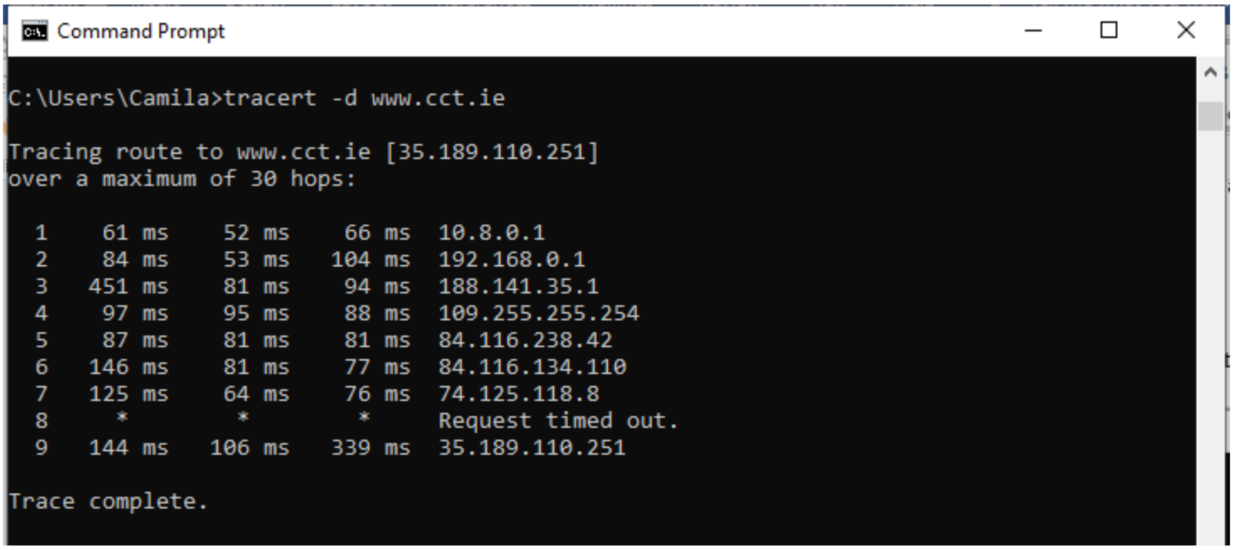
One of the advantages of using a VPN is that it helps to mitigate vulnerabilities like Man in the middle attacks (MITM). This is an attack which an intruder put itself in the middle of the communication between two parties who trust they are sharing information with each other. The objective of this action is to steal sensitive information, login certifications, card numbers, password, etc, for fraud purposes, irregular transactions or for the infiltration phase of other threats The attacker can control the communication traffic between the victims by reading, modifying, intercepting and changing it without leaving any trace and remaining hidden from the targets (Mallik et al, 2019).

The use of VPN helps to mitigate this kind of treat since it hides the header of the packets and the data. Even if the intruder tries to sniff the packets using Wireshark, it is not possible to see any information.

Testing VPN by using the traceroute command

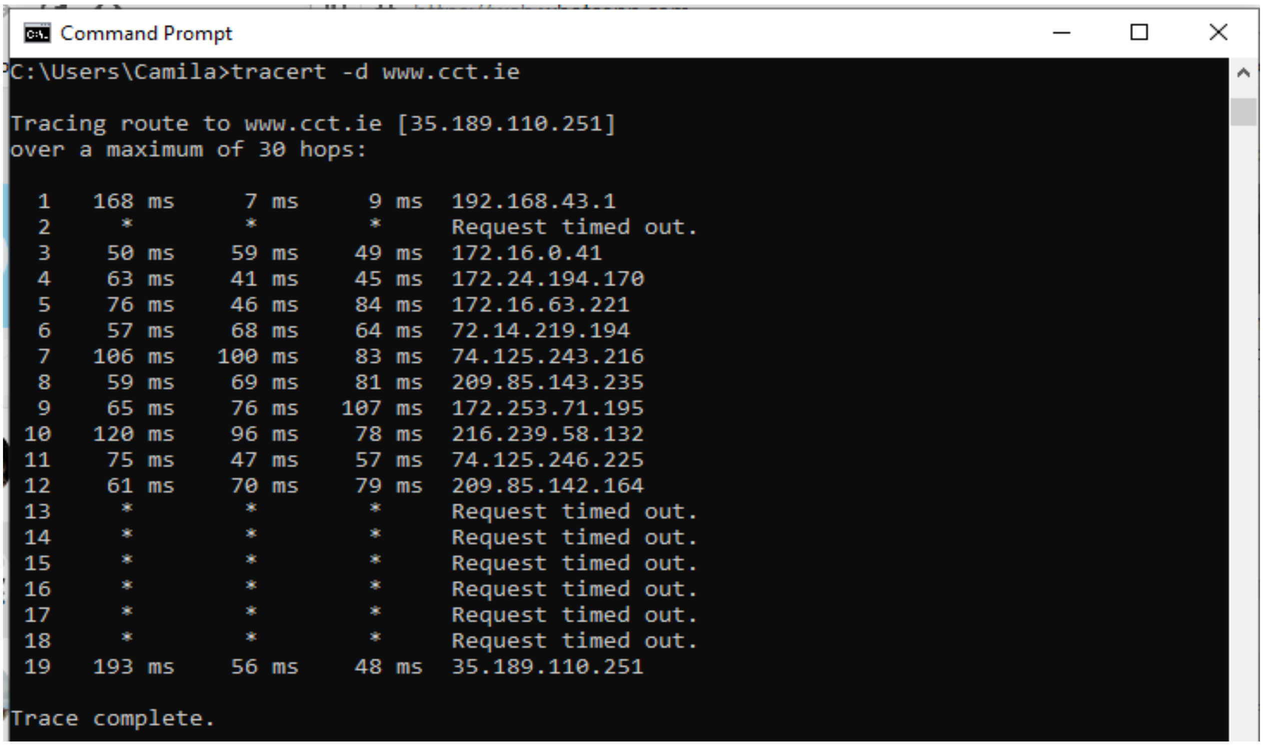
An alternative way of testing the usage of VPN is by using the command tracert -d which shows how the packets are being sent to one router to another. When the client OpenVPN is connected to the OpenVPN server all the traffic goes through the tunnel. For this reason, the tracert command shows that the fist hop is the OpenVPN server.

The screenshot bellow shows that the first hop in the trace is 10.8.0.1 which is the OpenVPN IP address and the second one is the default gateway of the home router 192.168.0.1 and finally goes to the Internet.



```
C:\Users\Camila>tracert -d www.cct.ie

Tracing route to www.cct.ie [35.189.110.251]
over a maximum of 30 hops:

  1    61 ms    52 ms    66 ms  10.8.0.1
  2    84 ms    53 ms   104 ms  192.168.0.1
  3   451 ms    81 ms    94 ms  188.141.35.1
  4    97 ms    95 ms    88 ms  109.255.255.254
  5    87 ms    81 ms    81 ms  84.116.238.42
  6   146 ms    81 ms    77 ms  84.116.134.110
  7   125 ms    64 ms    76 ms  74.125.118.8
  8     *        *        *     Request timed out.
  9   144 ms   106 ms   339 ms  35.189.110.251

Trace complete.
```

Meanwhile, without the use of OpenVPN server the first hop is the default gateway of the router provided by the ISP, in this case Three network.

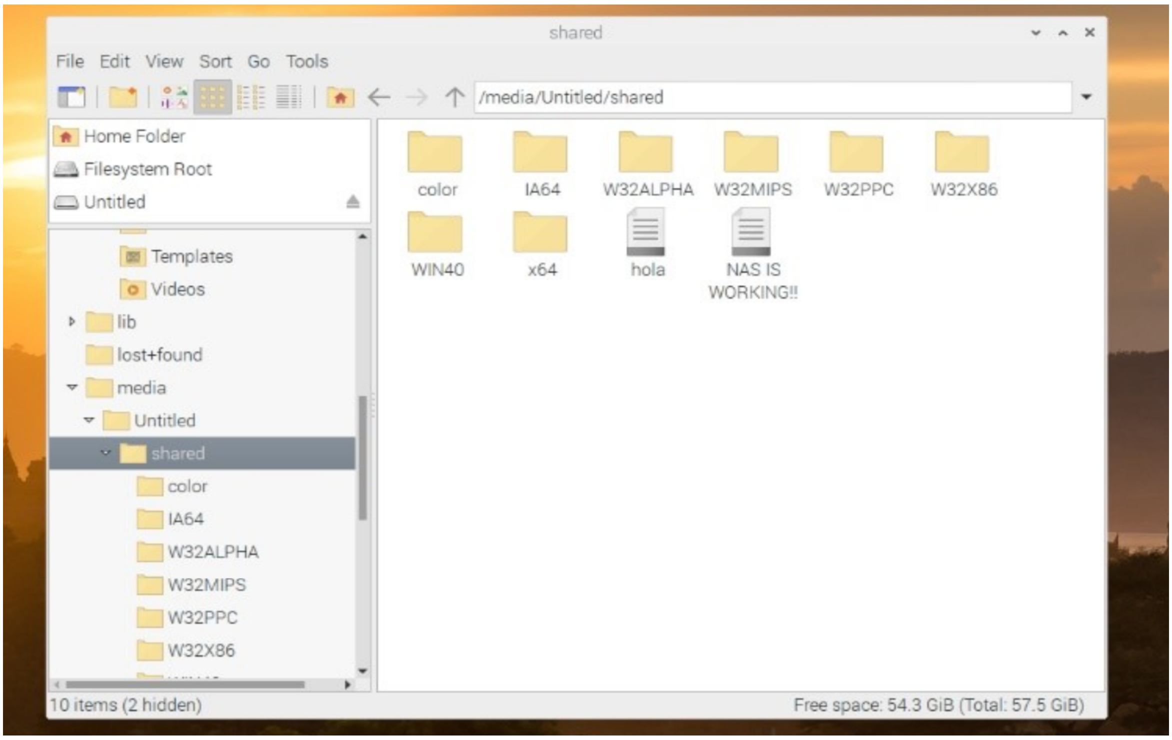


```
C:\Users\Camila>tracert -d www.cct.ie

Tracing route to www.cct.ie [35.189.110.251]
over a maximum of 30 hops:

  1   168 ms     7 ms     9 ms  192.168.43.1
  2     *        *        *     Request timed out.
  3    50 ms    59 ms    49 ms  172.16.0.41
  4    63 ms    41 ms    45 ms  172.24.194.170
  5    76 ms    46 ms    84 ms  172.16.63.221
  6    57 ms    68 ms    64 ms  72.14.219.194
  7   106 ms   100 ms    83 ms  74.125.243.216
  8    59 ms    69 ms    81 ms  209.85.143.235
  9    65 ms    76 ms   107 ms  172.253.71.195
 10   120 ms    96 ms    78 ms  216.239.58.132
 11    75 ms    47 ms    57 ms  74.125.246.225
 12    61 ms    70 ms    79 ms  209.85.142.164
 13     *        *        *     Request timed out.
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19   193 ms    56 ms    48 ms  35.189.110.251

Trace complete.
```

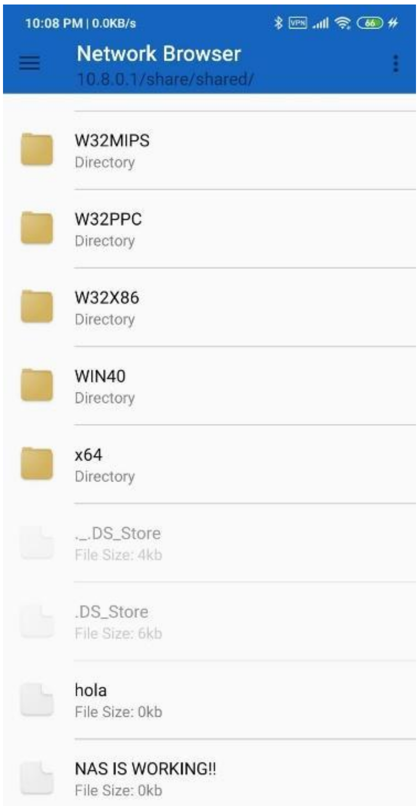
Testing Network-Attached Storage

The USB which stores the files and work as NAS in already plugged in the Raspberry Pi, so in order to test if the files can be accessed it is necessary to connect to the OpenVPN server again using the client OpenVPN software. Once the connection is established, it is possible to find the files inside the USB sticky and that it is connected to the Raspberry Pi.

The files can be accessed from the mobile phone and from the laptop.



The files can also be accessed from the mobile phone.



10:08 PM | 0.0KB/s

Network Browser

10.8.0.1/share/shared/

W32MIPS
Directory

W32PPC
Directory

W32X86
Directory

WIN40
Directory

x64
Directory

._.DS_Store
File Size: 4kb

.DS_Store
File Size: 6kb

hola
File Size: 0kb

NAS IS WORKING!!
File Size: 0kb

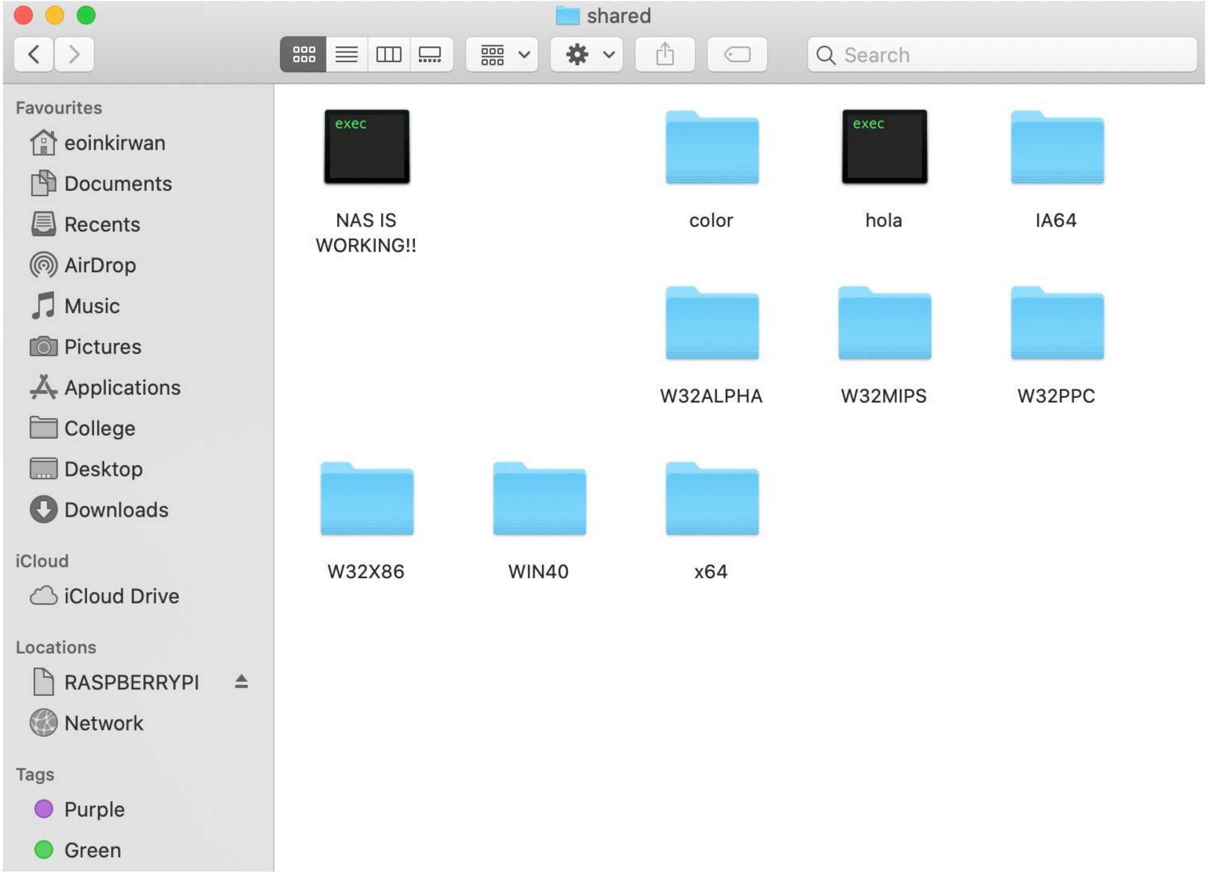


Figure showing the files that are stored in the USB being accessed from the laptop device.

Chapter 6 – Conclusion

This chapter concludes the whole process that started in Chapter 1 with the definition of the problem and objectives and goals. It discusses aspects of the implementation, testing and reflects about the learning outcomes from the process.

The whole idea for the project started with conversations about a trip that one the members of the group smart5 made to China on which he could not access his Facebook account due to the regulation in the country that forbids the access of contents from Google or other services.

The group had discussions about how to solve the matter by thinking about new technologies that could be created, but the idea of using a Virtual Private Network was the obvious answer for the question. However, implementing a VPN at home using a Raspberry Pi instead of using a service provider sounded interesting due to the challenge of the task, especially for IT students that are curious about how some technologies work and how they can implement their own technologies by themselves instead of using the ones available in the market.

The original goal of this project was to install the VPN server in the Raspberry Pi and connect the clients from everywhere in the world to their home network and using the Internet in a more secure way. However, after many sessions of discussions with the supervisor and suggestions about some features that could add more challenges and improve our main idea, the group decided to create a Network Storage Attachment system using the Raspberry Pi that could allow the user to have access to files and folder in the network.

Therefore, the group decide that the main goal of this project was to create a Virtual Private Network by installing an OpenVPN server in a Raspberry Pi and access files and folders remotely from another device from anywhere in the world using a public network.

The problem to be solved is the use of public networks in trips or public places, for example, airports and coffee places, when the user wants to keep their data secure and their information, such as location, safely while browsing on the Internet or doing important transactions on which their details need to be kept safe. Moreover, allow the user to access important files that are safely stored in their home network.

Plenty of research was done to obtain information about VPNs. The type of VPNs that exist and their protocols, how they work and the ports that need to be open to allow the traffic. The steps to install a VPN in a raspberry Pi, Cryptography, Port forwarding, dynamic DNS, etc. Besides, the group had to search for alternatives to overcome the challenges during the way. For example, the group researched about tools that could be used to access the device remotely in order to solve the problem of not having the monitor. While doing it, we came across with Putty software which allow the user to access the target device remotely in the same network the through the Secure Shell protocol.

One of the learning outcomes of this project certainly was to do the System Analyses and Design of our system. The members of the group had previous experience of doing analyses and design of software systems, but not experience of doing the analyses and design of a network system. For example, it was a big challenge to create diagrams to illustrate how the system work. Also, to decide which diagrams would be more appropriate to be used to show the process.

The learning outcomes include Cryptography. The group learned the difference between asymmetric, symmetric encryption and hash function. We have learnt the concept of RSA algorithm, Diffie-Hellman that is responsible to exchange the keys, SHA, that were used in the commands for the installation of the VPN. Knowing those algorithms, helped us to understand better how the Virtual Private Network and that the cryptography techniques are the main reason that makes a VPN be considered so secure. We have learned why there are two keys in the asymmetric encryption. The sender encrypts the message by using the public key and the receiver decrypts the message by using the private key.

The group also research about dynamic DNS and port forwarding which were essential concepts and configurations for the whole functionality of the OpenVPN. We came across to these terminologies while finding ways to connect the devices to the OpenVPN server. We had to configure our routers to accept a new rule and forward the traffic to and from the port 1194 to the Raspberry Pi where the OpenVPN server was installed.

Moreover, we had to use a dynamic DNS service, because the public IP address is dynamically assigned by the Internet Service Provider (ISP) and it can change anytime. By creating an account in No-IP website and register for the service, we could make sure that the domain name records are going to update itself if there is any change in the IP address.

The installation of Samba in order to enable Network-Attached Storage was also a good learning outcome. We learned the commands to install Samba and enable the sharing folder and we could test that it was working at the end.

Working with Linux was also a great learning outcome, because we learned how to work with the Linux shell and how to navigate through the directories creating files, folders, installing packets and editing files, etc.

For the testing part, the traffic was analysed by using Wireshark and it was possible to see that the packets from the VPN are secure due to the encryption. Sensitive data could not be seeing in the full stream, because of the use of the VPN. By doing the test, we could also discuss the advantages of using a VPN even if HTTPS is being used. HTTPS is already a protocol that encrypts the data; however, the VPN adds an extra layer of security to it, once it hides some information in the header, such as protocols that are being used, source and destination, etc.

Establishing the connection through the Virtual Private Network give us great ideas about how to expand the project for the future. A very good option would be to add smart devices that could be remotely accessed and controlled over the network. For example, a babysitter camera that could be turned on and off remotely in a secure way through the VPN. In the same direction, a food dispenser device that could be controlled to dispense food and water for pets while their owners are away. Those ideas are possible to be accomplish once we were able to access the resources of the network with the VPN.

This project turned out to be a good challenge and a good surprise for the group. At the beginning, we thought the task would be challenging, but we had no idea how much concepts we would have to research and learn in order to put everything in practise. Furthermore, we realised that we had to use concepts that we learned along the three years of College, for example, concepts from the modules of Data Communication, Network Forensics, Operating System, etc.

The learning outcomes not only are related to the technical aspects of the project, but put together an extensive work during a pandemic situation was a huge challenge. Dealing with the emotional pressure and with the new conditions of working from home, online meeting using Skype or Google Hangouts, not interacting face to face to discuss ideas, added up a bit more of a challenge to our original task.

At the end of the project, the group was really united working together and helping each other to finalise it. Maybe this is another learning outcome, in times like this days where people are feeling pressure due to the pandemic and insecure about their futures, the best thing is to keep close to their pairs, ask for help if they need and rely on people who are in the same situation trying to achieve a common goal. After all, we all have to keep going….

Appendices A- Project Planning

Appendices B – Group Journal

Date	Ideas/Actions	Next Actions
21/03/2020	Meeting between the members of the group to discuss the new actions for the semester: Feedback from the Faculty. The group agreed with some remarks about it, but disagreed about the planning part. The plan was made and the meetings were scheduled nearly every week to work on the project. The group feels that we need more meeting with the supervisor to show contents and have more feedbacks from him. The group agreed to register every meeting we will have over the semester and post it on Basecamps, so the supervisor and members of the Faculty can check our work. Also, the submissions needs to be submitted via Basecamp, so everyone will be aware of each other's work.	Camila will contact Ken to ask for revision about the marks. The next meeting will be scheduled after the feedback.
27/03/2020	Ken replied the email asking us to contact the supervisor about the marks and how to improve the project, however, he agreed to meet us to discuss some ideas. Jesus had an accident in Romania and does not know when he will be able to go back to Ireland. The group discussed about buying the Raspberry Pi, because had it with him at home.	Camila will send an email to Greg to schedule a meeting and contact Jesus to ask about his return to Ireland. Camila will research about Cryptography. Every member will research and watch tutorials about the installation of the VPN server on the Raspberry Pi.
28/03/2020	Meeting with the supervisor Greg: Discussion about the feedback from the Faculty group. The supervisor has told us that the VPN idea was too simple for what the college is expecting and that does not reflect how much we have been learning. Feedback about the report: the references were used many times, the planning was not clear enough because it did not outline all the tasks we are planning to do. Possibility of review the marks for planning criteria since the group claimed that we created a plan for the whole semester and we had meeting every week among us or with the supervisor.	Revision of the plan schedule in order to elaborate a new plan with more details. Research about the installation process and ways to improve the features of the VPN. Research more about technologies will be used that were missing from the first submission, for example, cryptography.

	Next actions will be taken: the installation process will start next week (04/03/2020). Jesus had an accident and is recovering from it. He will give the Raspberry Pi to us on the following week. The supervisor is aware of it. Conversation about working in group and letting the supervisor aware of any issues related to the team work, such as, submissions made after deadlines, priorities for the project since it has 20 credits, etc. Discussion about the possibility of the supervisor meet the group weekly to discuss the report and technical aspects of the project. The supervisor is not available for meetings every week due to his engagements with CCT. He agreed with being more in contact by emails. The supervisor will check the possibility of lending the monitor and the keyboard to the group for the installation process.	Meeting on 04/03/2020 to start the implementation of the VPN server on the Raspberry Pi.
04/03/2020	First meeting to start the implementation and installation of the VPN on the Raspberry Pi. Technical issues related to the monitor.	Agreed to meet up again on a close date to restart the process.
08/03/2020	Second meeting to start the installation process. Again, issue with the monitor. Division of tasks to be carried on.	Each member of the group will try to find tutorials that can be followed for the implementation of the VPN and the ones who have a monitor will implement that. New meeting will be scheduled to try it again as a group.
13/03/2020	Meeting online with the supervisor due to the closure of the college. The members suggested to create a connection site-to-site using two raspberry Pi and 3g connection. The supervisor instead, suggested to keep the original idea, but add new features, for example, access a webpage, access files and folders, etc.	Each member will research about new features to be implemented in the project, but also how to install the VPN.
18/03/2020	Meeting to try to install the VPN with all the members of the group. We were following a tutorial that it could work and help with the installation process.	Each member of the group will be responsible to research about methods of implementing the VPN.

	Issue with the tutorial. The files who were supposed to be generated weren't generated.	2 members will have a raspberry Pi and the rest will use a virtual machine to configure the commands.
25/03/2020	Conversation on WhatsApp about the next steps. Reginaldo is researching about the VPN protocols and port forward. Jesus is researching about Putty. Fernando is researching about Secure Shell. Camila is researching about Cryptography. Tenilde is researching about Wireshark.	All the members have individual topics to research about, but everybody will try to implement the VPN in the raspberry Pi.
01/04/2020	The VPN was installed on the raspberry Pi, but it is not possible to test it due router configuration. Virgin media does not allow the port forward to be enable manually, so Jesus have to contact the service provider to request it.	The members will try to install the VPN in their VMs and Raspberry Pi to get familiar with all the steps of the process.
06/04/2020	Online meeting with the members of the group to discuss the next steps. Division of tasks between two sub groups: 2 people will work on the implementation part and 3 people will work on the report. Discussions about System analyses and design about network projects. Which diagrams to use to represent our system? How to do the diagrams to illustrate a network?	Jesus will contact Virgin Media to ask for them to enable the port forward in his router. Camila will work in the implementation of the VPN again to get the screenshots to document the installation process.
09/04/2020	Online meeting with the supervisor. Update about the installation of the VPN and the next steps. Discussion about the system analyses and design. How to create diagrams for network projects.	Every member of the group will research about new features to improve the project, such as Network Attachment Storage, access a web page from VPN, access Virtual Machine.
22/04/2020	Online meeting with the members of the group to discuss the progress and the tasks every one was responsible for.	Division of tasks among the members of the group. One part will be responsible to work on the report, revising Chapter 1, and doing the System Analyses. Meanwhile, other members will work in the implementation and testing part.
24/04/2020	Online meeting with the supervisor to discuss the next steps of the project in regards to the testing using Wireshark. How to document the finding and proof all the effort the group is putting into the project.	The group divided the tasks that everybody would carry on and a new meeting will be scheduled soon. Jesus is testing VPN and implementing NAS.

	Doubts about how to perform the System Analyses and Design and elaborate diagrams that show the features of the system. Discussion about the new features will be implemented into our project, for example, access the website Conversation about deadlines and the screencast	
08/05/2020	Online meeting with the members of the group to discuss the next steps. Focus on the topics of that need to be included (research about NAS, TLS, implementation and, testing and diagrams). Reginaldo is doing the diagrams for the system analyses and desig, researching about TLS and port forwarding. Camila is documenting the implementation, testing and conclusion. Thenilde is updating the schedule. Fernando is implementing dynamic DNS on the raspberry Pi and researching about some topics of VPN (types).	Everybody agreed that the activities for the following days are focus on the report. Everybody will think about the topics to be included in the conclusion. How much we learnt while doing the projects and all the challenges we had. Meeting with Greg on Tuesday where we plan to discuss the screencast. From Tuesday to the submission date the group will work in the screencast and correct some aspects of the report.
12/05/2020	Online meeting with supervisor to discuss following steps regards to the presentation and final submission of the report. Tips about the analysis of the VPN using the Wireshark and how to put everything together in the report.	Jesus, Fernando, Thenilde and Reginaldo are going to work in the elaboration of screencast, meanwhile Camila will keep working in the report part. Camila will send the draft of the report to Greg for corrections and then update the documentation.

References

.

- Silberschatz, P. Galvin and G. Gagne, 2014. *Operating system concepts* (8th edition).
- Albarqi, A., Alzaid, E., Al Ghamdi, F., Asiri, S. and Kar, J. (2015) Public Key Infrastructure: A Survey.
- Android Authority (2018). *So what is a VPN, and why should you care? Android Authority.* Available in https://www.androidauthority.com/what-is-a-vpn-gary-explains-695574/. Accessed on 27/11/2019.
- Anicas, M. (2015). *What is a Firewall and How Does It Work?* [online] Digitalocean.com. Available at: https://www.digitalocean.com/community/tutorials/what-is-a-firewall-and-how-does-it-work.
- Avijit Mallik *et al.* (2019) 'Man-in-the-middle-attack: Understanding in simple words', *International Journal of Data and Network Science*, 3(2), pp. 77–92. doi: 10.5267/j.ijdns.2019.1.001.
- B, R., Foundation, R. and Foundation, R., 2020. Raspberry Pi 3 – Model B. [online] PiShop.us. Available at: <https://www.pishop.us/product/raspberry-pi-3-model-b-armv8-with-1g-ram/> [Accessed 12 April 2020].
- Barnes, R., 2017. *Samba: Set Up A Raspberry Pi As A File Server For Your Local Network — The Magpi Magazine*. [online] The MagPi magazine. Available at: <https://magpi.raspberrypi.org/articles/samba-file-server> [Accessed 01 May 2020].
- BISCHOFF, P., 2019. What Is Port Forwarding And Why Use It With Your VPN?. [online] Comparitech. Available at: <https://www.comparitech.com/blog/vpn-privacy/port-forwarding-vpn/#How_to_port_forward_on_a_VPN> [Accessed 9 May 2020].
- C. Negus (2015). *The Linux Bible. The comprehensive, tutorial resource*. Wiley ed. 9th ed.
- C. Scott, P. Wolfe, M. Erwin (1999). *Virtual Private Networks.* Why Build a Virtual Private Network. Chapter 1, p. 1-11. Ed. O'Reilly Media. 2nd edition.
- Commons.wikimedia.org. 2020. File:TLS Protocol Stack.Jpg - Wikimedia Commons. [online] Available at: <https://commons.wikimedia.org/wiki/File:TLS_protocol_stack.jpg> [Accessed 10 May 2020].
- Creativecommons.org. 2020. CC0. [online] Available at: <https://creativecommons.org/choose/zero/> [Accessed 7 May 2020].
- Crist, E. and Keijser, J., 2015. Mastering Openvpn. Packt Publishing.
- D. Athow. (2017). *Should you set your own VPN server? For the DYI's among us.* Techradar.pro. IT insights for business. Available on https://www.techradar.com/news/should-you-set-up-your-own-vpn-server. Access on 08/12/2019.
- DeMuro, J., 2020. *VPN Protocols And Which Is The Best To Use*. [online] TechRadar. Available at: <https://www.techradar.com/news/whats-the-best-vpn-protocol-to-use> [Accessed 12 April 2020].
- DNSimple, 2020. DNSimple support. DNS articles. DNSimple Corporation. Available at https://support.dnsimple.com/categories/dns/ [Accessed on 28/03/2020].

- E. Alvarez (2014). *Engadget. Sony Pictures Hack: the whole history*. Available on: https://www.engadget.com/2014/12/10/sony-pictures-hack-the-whole-story/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAL7K qnDPUElUZz6paAsfPmbq926nFXr56TcXJD3qHa2WlO6utIY-nIl5vVOk-tQ32Lmn8q0sjk9 9p2z-eG1F5qe1o47gNxEoHo8ACKueKuJdgdLbTpzUl06S3XI37In0HhgLsZkHvWEa71UDVTFCT8xnWlyw1d8ujCotLzX42B. Accessed on 27/11/2019.
- EasyRSA (2020). EasyRSA3. Available at https://easy-rsa.readthedocs.io/en/latest/ [Accessed on 15/05/2020].
- Ebrahim, M., 2017. *How To Use The Linux Samba Server - Dzone Performance*. [online] dzone.com. Available at: <https://dzone.com/articles/linux-samba-server> [Accessed 3 May 2020].
- Emmet, 2019. *Build Your Own Raspberry Pi NAS Server*. [online] Pi My Life Up. Available at: <https://pimylifeup.com/raspberry-pi-nas/> [Accessed 29 April 2020].
- ExpressVPN (2019). *ExpressVPN features.* Available on https://www.expressvpn.com/featu res. Accessed on 08/12/2019.
- Flack, R., 2020. *Network Attached Storage (NAS) Buying Guide - Newegg Insider*. [online] Newegg Insider. Available at: <https://www.newegg.com/insider/network-attached-storage-buying-guide/> [Accessed 10 May 2020].
- Guitteaud, K., 2015. *Ikev2 Site-To-Site VPN | SUPINFO, École Supérieure D'informatique*. [online] Supinfo.com. Available at: <https://www.supinfo.com/articles/single/244-ikev2-site-to-site-vpn> [Accessed 4 May 2020].Phillips, A., 2018. How To Turn Your Raspberry Pi Into A VPN Server Using Pi VPN. [online] Comparitech. Available at: <https://www.comparitech.com/blog/vpn-privacy/raspberry-pi-vpn/> [Accessed 7 May 2020].
- H. Bruijn, M. Jassen (2019). *Building cybersecurity awareness: The need for evidence-based framing strategies.* Available on file:///C:/Users/camil/Desktop/CCT/thirdYear2019/Cybs ersecurity/1-s2.0-S0740624X17300540-main.pdf. Accessed on 27/11/2019.
- Hirst, S., 2018. The Beginner'S Guide To VPN Port Forwarding. [online] Private Internet Access Blog. Available at: <https://www.privateinternetaccess.com/blog/the-beginners-guide-to-vpn-port-forwarding/> [Accessed 9 May 2020].
- Ibm.com. 2020. IBM Knowledge Center. [online] Available at: <https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_71/rzaiy/rzaiyl2tpsupport.htm> [Accessed 12 April 2020].
- IONOS Digitalguide. 2020. *What Is A Network Attached Storage (NAS)?*. [online] Available at: <https://www.ionos.com/digitalguide/server/know-how/what-is-a-network-attached-storage-nas/> [Accessed 5 May 2020].
- J. Rivington (2019). *The best VPN Service in 2019. Tom's Guide.* Available on: https://www.tomsguide.com/best-picks/best-vpn. Accessed on 08/12/2019.
- Jari, T., 2020. *INTRODUCTION TO SYSTEM ANALYSIS AND DESIGN*. [online] Available at: <https://www.academia.edu/5912421/INTRODUCTION_TO_SYSTEM_ANALYSIS_AND_DESIGN> [Accessed 13 May 2020].

- Jim Tomas, Howard Hooper. CCNP Security VPN 642-647 Official Cert Guide pg 283, 284
- *Journal of Information Security*, 6, 31-37. http://dx.doi.org/10.4236/jis.2015.61004
- K. Russel (2015). *How to set up your own Raspberry Pi powered VPN*. Technology. BBC news. Available in https://www.bbc.com/news/technology-33548728. Accessed on 03/12/2019.
- Kali by Offensive Security (2019). Available in https://www.kali.org/docs/introduction/what-is-kali-linux/. Accessed on 24/11/2019.
- LAKE, J., 2019. What Is TLS Encryption And How Does It Work? | Comparitech. [online] Comparitech. Available at: <https://www.comparitech.com/blog/information-security/tls-encryption/> [Accessed 7 May 2020].
- M. Eddy (2019). *What Is a VPN, and Why You Need One*. Available in https://uk.pcmag.com/features/88655/what-is-a-vpn-and-why-you-need-one. Accessed on 27/11/2019.
- Ma, E., 2019. [online] Systutorials.com. Available at: <https://www.systutorials.com/port-forwarding-using-iptables/> [Accessed 6 May 2020].
- Magalhaes, R., 2017. *Secure Socket Tunneling Protocol*. [online] TechGenix. Available at: <http://techgenix.com/secure-socket-tunneling-protocol/> [Accessed 1 May 2020].
- Negus, C. and Bresnaham, C., 2015. *Linux Bible*. 8th ed.
- Phillips, G., 2017. *The 5 Major VPN Protocols Explained*. [online] MakeUseOf. Available at: <https://www.makeuseof.com/tag/major-vpn-protocols-explained/> [Accessed 4 May 2020].
- R. Sharpe (YEAR). *Wireshark User's guide. Version 3.3.0.* Available in: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs. Accessed on 08/12/2019.
- Raspberry Pi Foundation (2019). Available in https://www.raspberrypi.org/. Accessed on 3/12/2019.
- Raspberrypi.org. 2015. Raspberry Pi 2 Model B. [online] Available at: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/> [Accessed 12 April 2020].
- Rouse, M., 2007. What Is Layer Two Tunneling Protocol (L2TP)? - Definition From Whatis.Com. [online] SearchNetworking. Available at: <https://searchnetworking.techtarget.com/definition/Layer-Two-Tunneling-Protocol-L2TP> [Accessed 12 April 2020].
- Rouse, M., 2020. *What Is Network Attached Storage? - Definition From Whatis.Com*. [online] SearchStorage. Available at: <https://searchstorage.techtarget.com/definition/network-attached-storage> [Accessed 4 May 2020].
- Salah-ddine, K. (2017). Overview Of Firewalls: Types And Policies Managing Windows Embedded Firewall Programmatically. Available at https://www.researchgate.net/publication/315614367_Overview_Of_Firewalls_Types_And_Policies_Managing_Windows_Embedded_Firewall_Programmatically/citation/download. Accessed on 18/0/5/2020.

- Schneier, B., n.d. [online] Schneier.com. Available at: <https://www.schneier.com/academic/paperfiles/paper-pptp.pdf> [Accessed 4 May 2020].
- Seagate.com. (n.d.). What is NAS (Network Attached Storage) and Why is NAS Important for Small Businesses? | Seagate US. [online] Available at: https://www.seagate.com/gb/en/tech-insights/what-is-nas-master-ti/ [Accessed 11 May 2020].
- smallstep.com. (2018). *Everything you should know about certificates and PKI but are too afraid to ask*. [online] Available at: https://smallstep.com/blog/everything-pki/ [Accessed 19 May 2020].
- Sohail, 2019. *What Is Samba Server And How To Setup Samba Server In Ubuntu Linux - Linuxandubuntu*. [online] LinuxAndUbuntu. Available at: <http://www.linuxandubuntu.com/home/what-is-samba-server-and-how-to-setup-samba-server-in-ubuntu-linux> [Accessed 5 May 2020].
- Tyson, J., Pollette, C., Crawford. S, (2020). How a VPN (Virtual Private Network) works. Remote Acces-VPN. HowStuffWorks. Available at https://computer.howstuffworks.com/vpn3.htm. Accessed on 28/03/2020.
- Vanparia, P., Ghodasara, Y., Donga, Mr (2003). Network Protocol Analyzer with Wireshark. Available at https://www.researchgate.net/publication/282385189_Network_Protocol_Analyzer_with_Wireshark/citation/download. Accessed on 21/04/2020.
- VPN One Click (2019). Available in https://www.vpnoneclick.com/types-of-vpn-and-types-of-vpn-protocols/. Accessed on 17/12/2019.
- VPNoverview.com. 2020. VPN Explained: How Does It Work? Why Would You Use It? | Vpnoverview. [online] Available at: <https://vpnoverview.com/vpn-information/what-is-a-vpn/> [Accessed 12 April 2020].
- What Is Transport Layer Security (TLS)?. 2020. Cloudflare. [online] Available at: <https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/> [Accessed 6 May 2020].
- Wilson, M (2019). SCP What is Secure Copy Protocol- definition and example. Available at https://www.pcwdld.com/what-is-scp. [Accessed on 10/05/2020].

Individual Contribution

Camila Pulz de Faria

- **First Semester**

For this project, my contribution started with my suggestions about the topic that we would choose. I had some ideas about websites or applications to offer services to different users, diary to register daily activities, etc.

I wanted to create a technology that could be used for a good cause in the community by improving the quality of life of people with disabilities or different conditions, for example, autism. However, I was also inclined to do something related to Network and Raspberry Pi, because it would be a different experience considering that we have been developing applications and websites for many modules over the course.

When Jesus joined the group and gave the idea of creating a device to be used on trips where some web contents are prohibited, for example, Facebook in China, I helped to look for more ideas that included Raspberry Pi. During a meeting with all members, we found an article on BBC website describing the process of installing the VPN in a Raspberry Pi, and then the group decided to implement it.

From the moment we agreed on the idea, I tried to plan the next meeting and following actions. For example, I would always text the other members on WhatsApp or approach them during the classes to suggest new actions or schedule new meetings. In addition, I was responsible to send most of the emails to the supervisor to schedule the appointments and take notes of the discussions we had during those meetings.

We divided the project in four main topics: Linux, VPN, Raspberry Pi and Cybersecurity. I was responsible to research about Linux. I went on different websites to ready about the Linux commands and I have bought two courses in Udemy (*Complete Linux Training Course to Get your dream job IT by Imran Afzal* and *Linux Operating System Fundamentals by Linux Academy*) to learn a bit more. I created a Virtual Machine on Oracle Virtual Box using a CENTOS distribution (free version of Red Hat distribution) and I have been studying and become familiar with the operating system since then.

I was responsible to write down the final report and put together all the references and research of each member, developing the idea and context for our project. To accomplish this task, I did some research about the other topics as well. I borrowed books in the CCT library about VPN and cybersecurity that helped me to elaborate some of the content of the report.

I helped to do the schedule for the first semester with the other members of the group. However, I did not participate on the elaboration of the plan for the second semester because I was working on the report.

I wrote down the method to install the VPN in the Raspberry Pi, but Greg said not to include this part on the first submission of the Chapter 1.

Once the report was finished, I formatted it. I created the summary, cover page and the structure of the file.

- **Second semester**

For the second semester, I was responsible to prepare the documentation about the project. I revised the Chapter 1 to see which contents should be included and if it was necessary to remove some of them.

I researched about Cryptography: asymmetric and symmetric cryptography and the examples of cryptographic algorithms. Also, I did some research about Public Key Infrastructure, EasyRSA, OpenVPN and Firewalls (UFW).

I helped my partners with the implementation of the VPN. I tried to find the tutorials for the installation and testing them to check if they would work for our project. At the end, when I finally found one which was successful, I used it and took the screenshots of the steps to add to our documentation. I also took some screenshots of the issues I had during the implementation in order to document the whole process.

I worked mostly in Chapter 4, 5 and 6 with the implementation part, testing and conclusion. I documented the whole implementation, since I was responsible for putting everything together in the report and discussed the challenges that we encountered during the process. I worked with Jesus in Chapter 5 (testing and evaluation) by adding some contents to his screenshots and I did the testing using the traceroute command. I also worked with Fernando for the testing part. We helped each other many times to troubleshoot errors in the connection with OpenVPN and to identify the connectivity between the client and the server.

I worked with Reginaldo to do the Design part. We discussed the information that would be relevant to be shown about our project and some of the contents that needed to be included.

I tried to organize the activities of the group, so I planned the online meetings with the supervisor and I suggested some of the meeting that we had among the members of the team. I registered the weekly activities and plan of actions to be submitted in the group journal and I tried to organize the activities and share the workload among the members of the group.

I formatted the report and created the Summary, Abstract and Acknowledgments section.

Fernando Aires

First semester

My contribution to this project started with the group definition, giving some suggestions about applications and contributing to the maturation of the main idea, suggesting project ideas for some social problems and community causes, such as recycling or better use of resources. However, we did not find a design idea that we could add to Raspberry Pi technology by creating something innovative; Therefore, after researching several options, we decided to adopt the idea of VPN as the main project.

I was responsible for VPN Security research, tried to familiarize myself with the topic by first going to some websites and then some specific books related the subject, mostly based on the content of specific websites because it was better explained and having more practical examples.

During the research, I tried to extract the most relevant content for the project, but found some difficulty with the VPN security topic once it is a very huge and multi-branch topic. Initially, I focused on data security and encryption, as they are the most cited topics on most websites and books, but knowing that there are several serious issues that we still need to talk in the future.

I collaborated with questions during meetings with the supervisor and with suggestions for drawing up the project plan diagram Jesus Colinas

Second semester

The second semester allowed me to better test my skills and deepen knowledge as it was a more practical part producing our prototype. I made some investments in equipment for testing such as monitor and HDMI cables.

I continued with the researches through books and videos related to security and tried to carry out tests and improvements to the project, but on the first attempts I was not succeed.

After deepened my research on areas such as VPN protocols, cryptography and how to improve system reliability with more secure protocols like Secure Shell (SSH) and SSL / TLS with books and technical materials in pdf to help the group with suggestions and basing my researchers on more reliable materials to expand our range of quality technical information in parallel as watching videos and reading about security topics in specialized blogs those methods were very important for me to simplify and better understand some concepts, and also collaborate with ideas to trespass the obstacles we were facing to the implementation of our project.

I also researched and executed the configuration of a DDNS (dynamic DNS) in the project to in order to update the IP address automatically in case the router was restarted or the ISP provider decided to change the IP, as it was not possible to do before we have the system running properly with all the steps of the implementation being done manually by command line. I performed and elaborated a sequence screenshots for the correct way to set up DDNS with all the commands lines until the on the raspberry pi server, as well as realizing tests on the laptop, smartphone and raspberry pi.

Another important participation was developing and editing the screencast of the project It allowed me to experiment and improve my knowledge in video edition.

The important aspects I participate in this project were solving some issues related to the Linux command lines on setting up a Virtual Private Network on a raspberry pi during the installation, configuration and testing the system that required the use of system analysis on Wireshark, contributing to the improvement to the security for the final result to the project.

I also participate and collaborate during all of the Skype meetings with the group and the our supervisor Greg.

Jesus Colinas

First Semester

For the project my contribution started by giving the idea about making a VPN with a Raspberry Pi. I explained to my team that the idea came from a trip that I made during the summer to Asia and I noticed that in some countries they censor the internet, for example social media, google maps and other navigation apps are forbidden. Primarily, the idea was to avoid the internet censorship, but when we researched about this VPN idea, we noticed that we could give it another important use that is data security for the users.

We talked about this idea with Greg and he thought that it would be a nice idea and we could give it a go.

We decided as a team to officially make it our proposal idea for this project. We divided the research between, VPN, raspberry Pi, Linux and Cybersecurity.

I primarily focused on the VPN research. I researched everything about VPNs, for example, what it is, what it does, the history of VPN and how it works. When I did this VPN research, I came across that some other information that I did not understand how they worked and I had to extend my research because they are related with VPN for example encryption. I found out that there is different types of encryption that it could be used for a VPN for example 256 bit encryption and 128 bit encryption. Other areas that I researched was OpenVPN and PPT (Point to point tunnelling protocol) these parts were not included in the report because we will not be using PPT but I had to research about them to be able to have a better understanding about VPNs. I researched about how to implement this idea and we found this BBC website that explains how to implement it. It also explains the commands that are required to be used for making our own VPN with a Raspberry Pi.

I did a small research about cryptography that we will talk more about it next semester

I helped to elaborate the schedule for the first semester, everybody helped to do the first semester schedule.

I made the Use case diagram and the architectural diagram that we added in the appendices because Greg told us that is something that we will need for next semester. More explanations about these diagrams will be given next semester.

Second semester

I kept doing my research about how to implement our VPN into the raspberry pi, we found some tutorials and some recommendations, but I tried to implement them but most of them did not work properly. Mostly because they were not complete, or the commands did not work. I did a small personal investment and I bought a Raspberry Pi 4. I did this because it is a lot faster, it has faster processor, more ram and wi-fi chip is better also because we are 5 in the group and we had just one, so I gave the raspberry pi to another person in my team to keep trying to implement the VPN.

After a lot of research, I was almost able to implement the VPN. I set a static IP address on the Raspberry PI. I assigned the port 1194 to redirect the traffic as Open VPN uses that port. I set the 256-bit encryption for my key. I do not have a static IP address and it changes so often so I had to set up my own DNS server on noip.com and I called the DNS service Smart5VPN. I also downloaded the tool from noipaddress.com to update my home's public address to my DNS server in case it changes like that the VPN will continue working without having to configure the IP address again. I created a username, password and certificate. Once all the VPN setup was done, I tried to connect to it but I couldn't as my internet service provider blocks the port forwarding. I had to call my ISP and ask them to enable it. They had to change my IPV6 to IPV4 to be able to enable port forwarding. I created a port forwarding rule in my modem with my Raspberry Pi's IP address. I tested it with the Open VPN app, set my certificate and password and I finally was connected to my VPN.

Camila was able to find another way to setup the VPN and we decided to use Camila's solution for our documentation as it was more suitable for our project.

I did some research about how to use WireShark and how to look at the traffic and packets so I can be able to do the tests.

I did some testing for our VPN using WireShark. I tested that it was working checking how the public and dynamic IP address change because once you connect to the VPN it gives my home's public IP address as the VPN redirects all the traffic to my home's internet. I also tested and took screen shots of how the VPN encrypts the traffic on a HTTP website as it uses an unsecure protocol. Once connected to the VPN I did the same test using the HTTP website and successfully it encrypts the traffic and I wasn't able to track the source as the VPM masks it. I also took screen shots of how the encryption handshake happens when I connect my device to my VPN.

I researched about NAS (Network-Attached Storage) and how to implement it on the Raspberry Pi. After a few attempts, I was able to implement NAS using Samba. I did some research about Samba and NAS and I highlighted the most important parts with the references to add it to our report.

I did some testing for the NAS showing how it works in my network. I tested it on different devices like my cell phone and other laptops and I was able to see the USB folder working as NAS. I took screen shots showing how I was able to save files and download them in another device using my VPN. I also took screen shots explaining each step of how to implement NAS in Linux.

I also participated in all the meetings that took place with our supervisor Greg and all the group meetings that we had on Skype.

Reginaldo Pereira dos Santos

- ***First semester***

For this project I have attempt in our weeks group meeting for discussion, planning and elaborating the next steps that should be done by the group. In one of these meeting I have come up it the idea where it was in building a Recycle App that users could use their phone to measure the amount of recycle garbage produced per day and per months and it could by the end generate a analysed graphic. This idea have a great accept by the group, but we have decided go for other idea less complicated.

After project definition, we have decided to do meetings to open a discussion about the VPN and Raspberry Pi to became a Final Project. Once everyone agreed on that, we have pointed to do some research, starting the project. I had decided go for Linux and VPN at the beginner, while its have done I have decided by myself go a little further in security and raspberry.

In Linux research I have read many article about, going further in Linux commands in general. While I've doing research I decided to download and install a Linux OS in my laptop. As we had Linux Server experience in class I decided install one version of Linux OS with interface. That gave me possibility in getting know Linux commands and familiarize with Linux interfaces.

After some test and time using it, I really enjoyed it by the end getting install and use a few software installed in it.

Researching VPN basic were necessary to know the virtual private network. This research gave me idea and information to start a introduction discussions and start to go further in this topic, while out classmate Jesus has been pointed be responsible for VPN researches.

Linux OS and Linux Server: as I've mentioned above, after some deep researches, using a Virtual Machine I have installed in my laptop bolt of Linux. I have done some test using Linux command and tried to get more familied with Linux commands.

VPN and Raspberry Pi : this part of project research consisted in watching YouTube videos that could clarify the doubts and help us with information to build our private VPN. On internet we've have seem some of videos in " how to build your own VPN".

Kali Linux: after supervisor meeting, he had mentioned about the Linux kali for set up a VPN and Raspberry and then I have done some research about it too. One of articles that I have read definitely show us that Linux most will be used to set up our project.

Types of VPN and VPN Provider: analysing the VPN provider while the kind of service are offered to the customers.

Introductions discussions and helps as Camila wrote it down.

Project schedule plan discussed in group, by the end I have made same change on that and make it more easily to understand and see the steps, everybody has been agreed on that way.

Project schedule second semester design and discussed with the group.

Schedule final with data and design.

Contributions with project name.

- **Second semester**

My contribution to this project on second semester started with the group definition steps on- line meetings.

As the beginning I have been helped Camila with the report. Its involve articles researches and resume. While the project started growing in material researches and practical parts I have sprit out my tasks in different directions while the group needed.

Raspberry Pi Manual : I have helped with the user guide as it has been solicited to be made.
Those tasks demanded were:

System analyses project: it has been demand creation and solutions for Diagrams. After the diagrams has done, some researches about it has made necessary, while diagrams demand more explanations.

After the project diagram decided, I have to use a correct tolls to made it, and use specific software to draw down those.

Protocols: A part of diagrams, I have to research some Protocols to the project report. It's basically involve, researches for articles, analyse the points and wrote down the necessary arguments about it.
OpenVPN
Layer 2 Tunnelling Protocol
Point-to-point Tunnelling Protocol (PPTP)
Transport Layer Security (TLS)

Tenilde Borges

- **First semester**

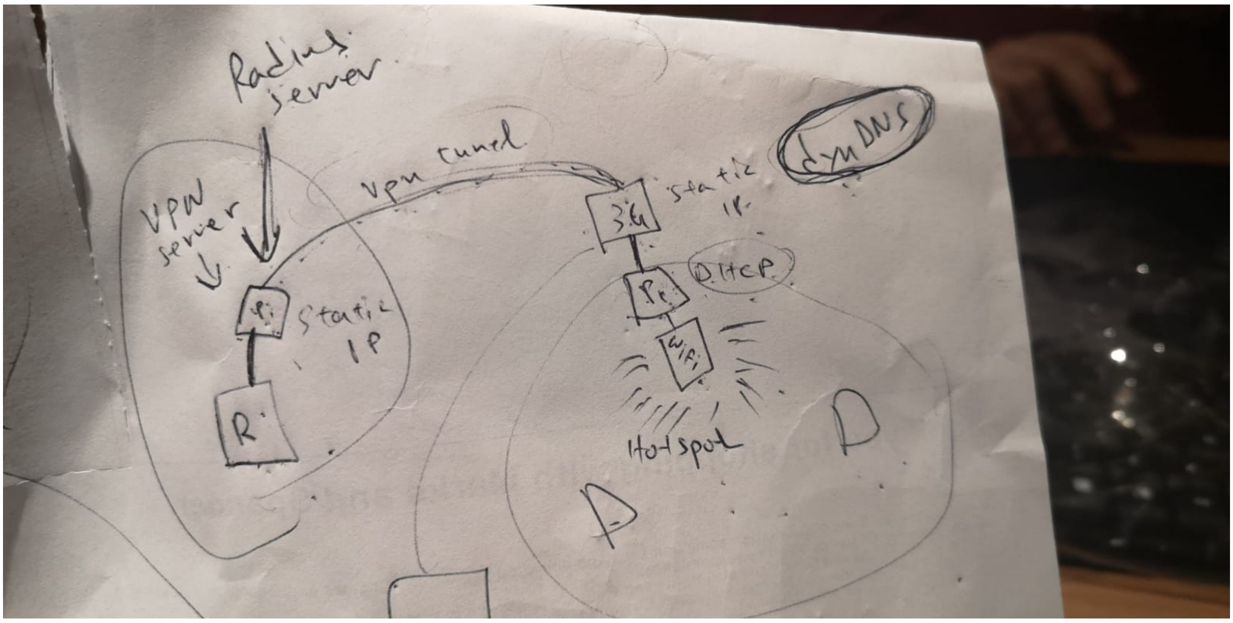
The contribution that I did in this first part for this project was with my searches and attending of all meetings. In the beginning everybody had a mission to come up with ideas so that we start to do the project.

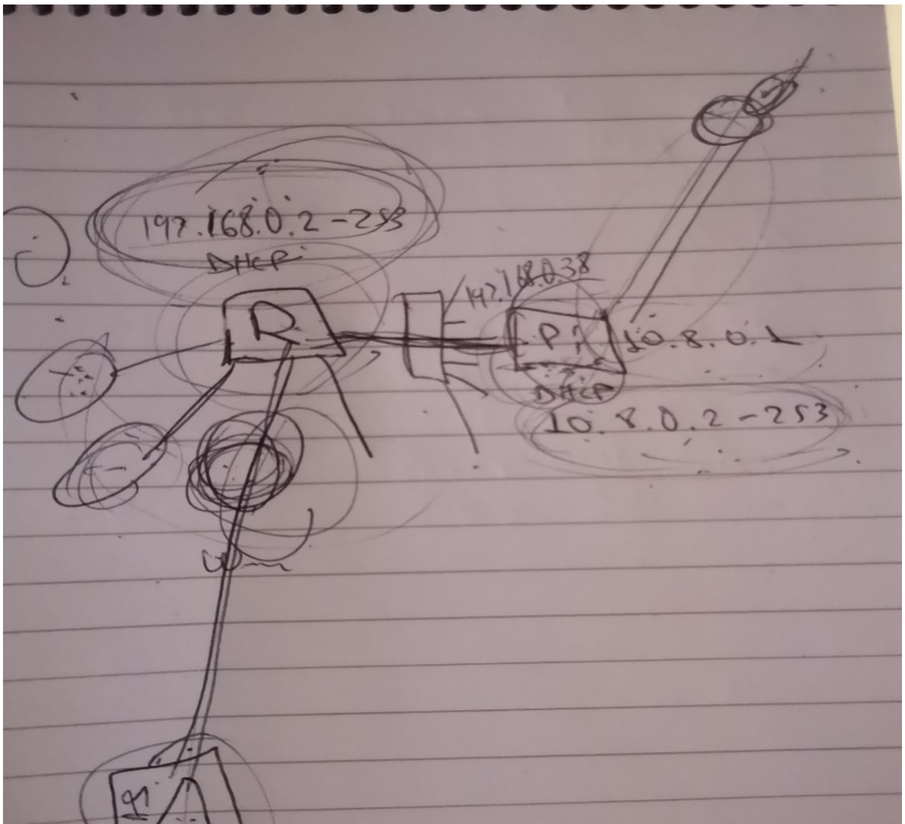
I had done a game project before and a member of group suggested that we could develop a game based on my project, to help people of some deficiency, but as many good ideas has come up me and all group decided placing in practice the Jesus idea. He voiced his ideas about create a security internet by using VPN and explained us why we should do a VPN project. As soon we decided what we want to make up everybody has got responsible to do something. I had the responsibility to search about Raspberry Pi. I searched in the internet everything that I needed to know about Raspberry Pi. I read some article in Portuguese as well to understand more how Raspberry Pi has become a such important technology nowadays. So I tried doing my best in first part and I hope to contribute more and more and second part of this project.

- **Second semester**

My contribution in the second part of the project was participating in the meetings we had to define points that needed to be agreed in order to move forward on the project, such as defining the tasks that each co-operator will have to do. I gave my best with my contributions in everything I was asked to do. The project was very intense and difficult to do due to the fact that we only have the internet as a source of resources, in this second part we did not have much chance of using the books that could support in our researches. I participated in the research that was necessary to improve the VPN As well as how I was covered up to improve the project scheduler and also to do a part of the screencast. In my screencast I talked about the VPN protocols and how they work to offer different features and also level of security .So this was my participation this project.

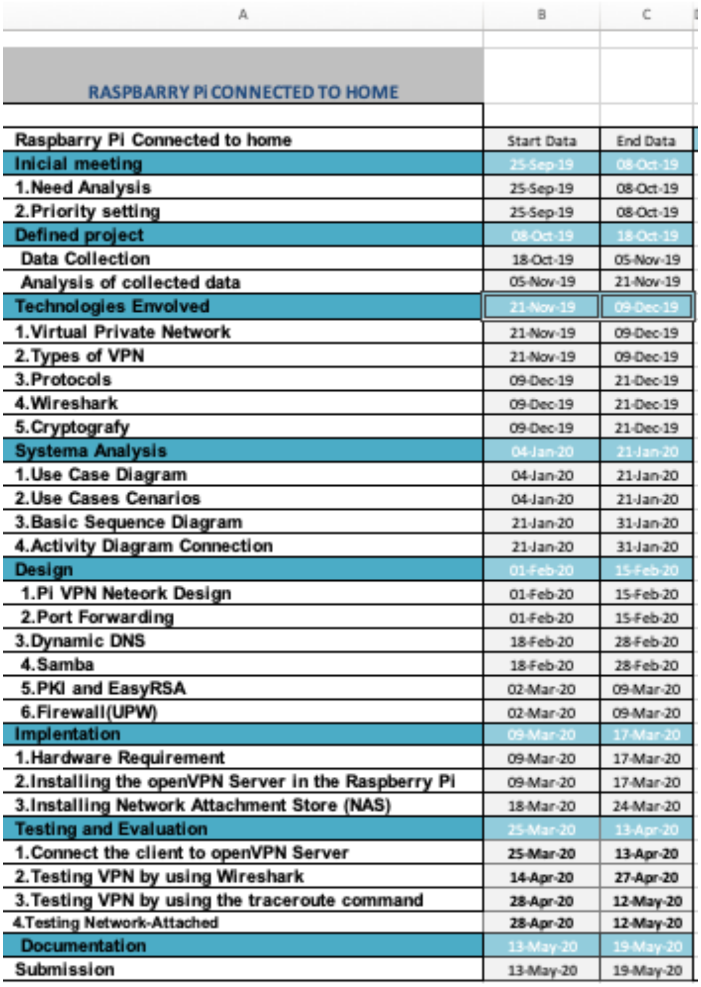
Appendices C – Photos and notes



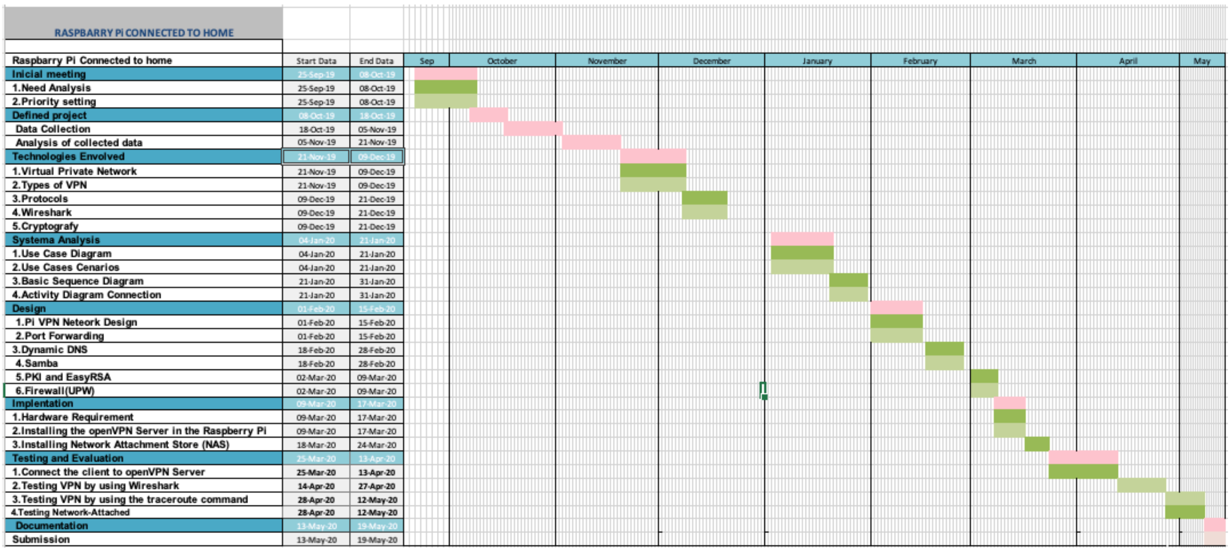


3665

CORSAIR



A	B	C
RASPBARRY Pi CONNECTED TO HOME		
Raspbarry Pi Connected to home	Start Data	End Data
Inicial meeting	25-Sep-19	08-Oct-19
1.Need Analysis	25-Sep-19	08-Oct-19
2.Priority setting	25-Sep-19	08-Oct-19
Defined project	08-Oct-19	18-Oct-19
Data Collection	18-Oct-19	05-Nov-19
Analysis of collected data	05-Nov-19	21-Nov-19
Technologies Envolved	21-Nov-19	09-Dec-19
1.Virtual Private Network	21-Nov-19	09-Dec-19
2.Types of VPN	21-Nov-19	09-Dec-19
3.Protocols	09-Dec-19	21-Dec-19
4.Wireshark	09-Dec-19	21-Dec-19
5.Cryptografy	09-Dec-19	21-Dec-19
Systema Analysis	04-Jan-20	21-Jan-20
1.Use Case Diagram	04-Jan-20	21-Jan-20
2.Use Cases Cenarios	04-Jan-20	21-Jan-20
3.Basic Sequence Diagram	21-Jan-20	31-Jan-20
4.Activity Diagram Connection	21-Jan-20	31-Jan-20
Design	01-Feb-20	15-Feb-20
1.Pi VPN Neteork Design	01-Feb-20	15-Feb-20
2.Port Forwarding	01-Feb-20	15-Feb-20
3.Dynamic DNS	18-Feb-20	28-Feb-20
4.Samba	18-Feb-20	28-Feb-20
5.PKI and EasyRSA	02-Mar-20	09-Mar-20
6.Firewall(UPW)	02-Mar-20	09-Mar-20
Implentation	09-Mar-20	17-Mar-20
1.Hardware Requirement	09-Mar-20	17-Mar-20
2.Installing the openVPN Server in the Raspberry Pi	09-Mar-20	17-Mar-20
3.Installing Network Attachment Store (NAS)	18-Mar-20	24-Mar-20
Testing and Evaluation	25-Mar-20	13-Apr-20
1.Connect the client to openVPN Server	25-Mar-20	13-Apr-20
2.Testing VPN by using Wireshark	14-Apr-20	27-Apr-20
3.Testing VPN by using the traceroute command	28-Apr-20	12-May-20
4.Testing Network-Attached	28-Apr-20	12-May-20
Documentation	13-May-20	19-May-20
Submission	13-May-20	19-May-20



RASPBARRY Pi CONNECTED TO HOME		
Raspbarry Pi Connected to home	Start Data	End Data
Inicial meeting	25-Sep-19	08-Oct-19
1.Need Analysis	25-Sep-19	08-Oct-19
2.Priority setting	25-Sep-19	08-Oct-19
Defined project	08-Oct-19	18-Oct-19
Data Collection	18-Oct-19	05-Nov-19
Analysis of collected data	05-Nov-19	21-Nov-19
Technologies Envolved	21-Nov-19	09-Dec-19
1.Virtual Private Network	21-Nov-19	09-Dec-19
2.Types of VPN	21-Nov-19	09-Dec-19
3.Protocols	09-Dec-19	21-Dec-19
4.Wireshark	09-Dec-19	21-Dec-19
5.Cryptografy	09-Dec-19	21-Dec-19
Systema Analysis	04-Jan-20	21-Jan-20
1.Use Case Diagram	04-Jan-20	21-Jan-20
2.Use Cases Cenarios	04-Jan-20	21-Jan-20
3.Basic Sequence Diagram	21-Jan-20	31-Jan-20
4.Activity Diagram Connection	21-Jan-20	31-Jan-20
Design	01-Feb-20	15-Feb-20
1.Pi VPN Neteork Design	01-Feb-20	15-Feb-20
2.Port Forwarding	01-Feb-20	15-Feb-20
3.Dynamic DNS	18-Feb-20	28-Feb-20
4.Samba	18-Feb-20	28-Feb-20
5.PKI and EasyRSA	02-Mar-20	09-Mar-20
6.Firewall(UPW)	02-Mar-20	09-Mar-20
Implentation	09-Mar-20	17-Mar-20
1.Hardware Requirement	09-Mar-20	17-Mar-20
2.Installing the openVPN Server in the Raspberry Pi	09-Mar-20	17-Mar-20
3.Installing Network Attachment Store (NAS)	18-Mar-20	24-Mar-20
Testing and Evaluation	25-Mar-20	13-Apr-20
1.Connect the client to openVPN Server	25-Mar-20	13-Apr-20
2.Testing VPN by using Wireshark	14-Apr-20	27-Apr-20
3.Testing VPN by using the traceroute command	28-Apr-20	12-May-20
4.Testing Network-Attached	28-Apr-20	12-May-20
Documentation	13-May-20	19-May-20
Submission	13-May-20	19-May-20