

Post-Quantum Cryptography VPN Application: Python Implementation

COMP8047 – Major Project

Kuan-Yu (Gray) Chen – A01088589
10-30-2023

Table of Contents

1.	Introduction	3
1.1.	Student Background.....	3
1.1.1.	Education	3
1.1.2.	Projects	3
1.2.	Project Description.....	4
1.2.1.	Essential Problems	4
1.2.2.	Goals and Objectives.....	5
2.	Body	5
2.1.	Background	5
2.2.	Project Statement.....	7
2.3.	Possible Alternative Solutions.....	7
2.4.	Chosen Solution	8
2.5.	Details of Design and Development.....	8
2.5.1.	Deliverables.....	8
2.5.2.	Milestones.....	9
2.5.2.1.	Milestone 1: VPN with OpenVPN Protocol	9
2.5.2.2.	Milestone 2: Post Quantum Algorithm Implementation	10
2.5.2.3.	Milestone 3: GUI Integration	13
2.5.3.	Installation Manuals.....	14
2.5.4.	Setup Guide.....	24
2.5.4.1.	Setup Script	24
2.5.4.2.	Port Forwarding on Home Router (Server only)	25
2.5.5.	User Manuals	28
2.5.5.1.	VPN application (OpenVPN Protocol)	28
2.5.6.	System Diagram (VPN Application).....	38
2.5.7.	Software Architecture Diagram (VPN Application).....	39
2.5.8.	System Use Case Diagram.....	42
2.5.9.	Network Diagram of the VPN Application	43
2.5.10.	GUI UML Diagram	43
2.6.	Testing Details and Results	45
2.6.1.	VPN with OpenVPN Protocol	45
2.6.2.	Post Quantum Algorithm Implementation to VPN	52

2.6.3. GUI Integration.....	59
2.7. Implications of Implementation.....	64
2.8. Innovation.....	65
2.9. Complexity	66
2.10. Research in New Technologies	66
2.11. Future Enhancements	68
2.12. Timeline and Milestones.....	69
3. Conclusion.....	71
3.1. Lessons Learned.....	71
3.2. Closing Remarks	72
4. Appendix	73
4.1. Approved Proposal.....	73
4.2. Project Supervisor Approvals.....	73
5. References	74
6. Change Log	75

1. Introduction

1.1. Student Background

I am an international student from Taiwan and am currently in the Bachelor of Technology in the Computer Systems program. Before joining the bachelor's program, I completed the Computer Systems Technology Diploma and have two years of programming experience. Currently, I specialize in network security applications development, and I will be using those experiences and knowledge to take advantage of learning new application and encryption protocols.

1.1.1. Education

British Columbia Institute of Technology

- Bachelor of Science, Applied Computer Science
Network Security Applications Development Option Sep 2021 – Current
- Computer Systems Technology Diploma – Web and Mobile Jan 2019 – Dec 2020

University of British Columbia

- Bachelor of Science – Applied Biology Sep 2017 – Apr 2018

1.1.2. Projects

- Search Engine Website for Recycling for City of Vancouver, BCIT Jan 2019 – April 2019
- Mobile-friendly Web Game, BCIT Apr 2019 – May 2019
- Cross-Platform Mobile Application for Voting, BCITSA Apr 2020 – May 2020
- Mobile Application for Construction, BICC Professionals Sep 2020 – Dec 2020

1.2. Project Description

Post-quantum cryptography, which can be referred to as quantum-resistant cryptography or quantum-safe cryptography, aims to construct public key cryptosystems that remain secure from attackers with quantum computers. This project's goal is to create a VPN application that integrates post-quantum cryptography with the **OpenVPN** protocol. By implementing this application, users will be able to use the internet with their data encrypted and secured even after quantum computers are built. This application will consist of two parts: VPN Client and VPN Server. The VPN Client will encrypt the data from the user by applying CRYSTALS-KYBER algorithm and send the traffic through the VPN Tunnel to the VPN Server. On the other side, the VPN Server will listen to communications on specified port to receive data and decrypt the received data. It will also forward all traffic requests from the client to the internet with encryption while hiding the client's address and geolocation.

1.2.1. Essential Problems

In the development of this Quantum-Resistant VPN project, essential challenges are being addressed to ensure the security and functionality of the VPN service in an era where quantum computing poses a potential threat to conventional cryptographic systems. Key issues include the selection and implementation of quantum-resistant cryptographic algorithms for secure key exchange, data encryption, and digital signatures. A focus is placed on creating a robust key exchange mechanism that prevents the exposure of shared secrets to quantum attacks, considering quantum key distribution as a potential solution. The project also evaluates the VPN's security against quantum attacks, actively seeking to remain secure even when faced with the advent of quantum computing. These challenges form the core of the efforts in building this Quantum-Resistant VPN project that meets the growing demand for secure, quantum-resistant communication.

1.2.2. Goals and Objectives

The problem my application will solve is that users do not have to worry about the security of their data anymore, even when quantum computers are built. With the combination of the current encryption protocol and post-quantum cryptography, the application not only secures the user's data from classical computers but also prevents future threats when quantum computers are built and widely used by attackers. By implementing post-quantum cryptography into my application, the VPN application will allow users to keep their data secure against both quantum and classical computers. The protocol I will use to develop my VPN application is the OpenVPN protocol, which is the one of the most widely used protocols for VPN applications. The algorithm I will use for implementation is the CRYSTALS-KYBER Algorithm, which is one of the algorithms that are selected by the NIST (National Institute of Standards and Technology) for Post-Quantum Cryptography in 2022.

2. Body

2.1. Background

As the internet grows, companies start to move their businesses online, allowing customers to do transactions online. However, websites become vulnerable once their confidential information, such as credit card numbers, passwords, or social insurance numbers, are leaked to cyber criminals. To prevent cyber-attacks, cryptography plays an essential and irreplaceable role in the security of all internet communications.

Nowadays, public key cryptography is widely used for securing data. The most common example of public key cryptography is the security of "HTTPS" (Hypertext Transfer Protocol Secure) web pages. When the customers are making the transactions through the web page, Transport Layer Security (TLS) or Secure Sockets Layer (SSL) protocols are applied to encrypt the exchanged data with their own

algorithms. However, all the public key algorithms are based on complicated mathematical problems, such as discrete logarithms or prime factors of a composite number. Currently, those mathematical problems are almost impossible for classical computers to solve. This means that if a technology is developed to solve those mathematical problems, the data that are encrypted with the problems will no longer be safe.

Currently, scientists and engineers use supercomputers to solve difficult problems. However, most of the supercomputers are very large. They are built with thousands of classical CPU and GPU cores, which makes them so longer “super”. Also, if the problem has a high degree of complexity, such as containing a very large number of variables, supercomputer will often fail to solve, which makes it no longer “super”. The technology that is rising for solving such high complexity problems is called quantum computing. Quantum computing is a rapid-emerging technology that is developed to solve problems that are too complex for classical computers, including supercomputers. The computers that are built based on this technology are called quantum computers.

Fortunately, quantum computing is still a fairly new technology. Therefore, quantum computers have not existed and have not been built in the world yet. However, by the time quantum computers are built, they can easily break most of the problems that are generated by the current encryption algorithms. As a result, all data that is encrypted with the current encryption algorithms is no longer safe. To prevent this happening, quantum-safe cryptography emerged. Quantum-safe cryptography aims to develop algorithms that are safe enough to prevent attacks by both classical and quantum computers. One way of the implementation is to use mathematical techniques such as error correcting codes, lattices, or multivariate equations to develop quantum-safe cryptography (Open Quantum Safe, 2022).

2.2. Project Statement

This project is to develop a quantum resistance VPN application by using the quantum resistance algorithms provided by the Open Quantum Safe Project with the OpenVPN Protocol.

2.3. Possible Alternative Solutions

There are some possible alternative solutions to the project based on the following categories:

- **VPN Protocol**

There are many selections for VPN Protocol, for example OpenVPN, IKEv2/IPSec, WireGuard, and others. Each protocol serves different purposes and has different security measures.

- **Programming Languages**

There are many programming languages that we can choose from, including Python, C, C++, Java, and other languages. Each of the languages has their advantages and disadvantages.

- **Quantum Resistance Algorithms**

According to the Open Quantum Safe (OQS) Project, there are many algorithms that are provided. It is divided into two parts: KEMs (Key encapsulation mechanisms) and Signature schemes. Some of the algorithms provided by OQS project include Kyber, McEliece, Dilithium, and Falcon.

2.4. Chosen Solution

Based on the alternative solutions above, it was decided to use the CRYSTALS-KYBER algorithm to develop a VPN Application with OpenVPN Protocol using Python. The reason for choosing the CRYSTALS-KYBER algorithm is because it is the most recent algorithm that NIST (National Institute of Standards and Technology) had selected for KEMs (Key encapsulation mechanisms). OpenVPN Protocol was selected because it is one of the most widely used VPN Protocols in the world.

2.5. Details of Design and Development

2.5.1. Deliverables

- **VPN Client**

A client application is delivered with sample configuration and key files. The program is built in a simple user interface by Python.

- **VPN Server**

A server application is delivered with sample configuration and key files. The program is built in a simple user interface by Python.

- **Setup Script**

A setup script is delivered to help the user automate the setup process instead of manually installing the required libraries and software.

- **Final Report**

A detailed report that contains all necessary information about the project is delivered.

2.5.2. Milestones

Each of the following subsections discusses the details of each milestone during the implementation of the project.

2.5.2.1. Milestone 1: VPN with OpenVPN Protocol

As I embarked on Milestone 1 of setting up a VPN using the OpenVPN Protocol, I initially felt quite lost. Understanding the protocol turned out to be more challenging and time-consuming than I had expected. However, I was determined to figure things out. One of the first problems I had to overcome was installing all the necessary software and configuring both the client and server components. I ran into the problem of being unable to access any websites on the client side (didn't get the server's response back) after connecting to the server. After carefully investigating the packet captures, the problem is solved by setting up iptables rules on the server side to forward all traffic from the client. This was a significant step, marking my progress in learning how the VPN worked. But my journey was far from over.

I soon encountered a new problem: I couldn't get the client and server to connect when they were on different networks with their own unique public IP addresses. After some careful investigation, I found the solution. It turned out that I needed to make a few adjustments to my home router. By setting up port forwarding on my router, I allowed the server to communicate beyond my local network. This was a critical step because it meant that the server could now reach out to the wider internet, connecting to the client and making the VPN functional.

Milestone 1 was a learning experience that not only tested my technical skills but also taught me the importance of paying attention to detail when working with VPNs and networks. It reminded me that even the smallest adjustments can have a big impact on achieving a successful connection.

2.5.2.2. Milestone 2: Post Quantum Algorithm Implementation

While exploring the quantum resistance algorithms from the Open Quantum Safe project, I discovered that the project divided the algorithms into two parts: KEMs (Key encapsulation mechanism) and Signature schemes. After spending time understanding how the algorithm and the OpenVPN protocol works, I first ran into problems when applying Kyber with the OpenVPN Protocol.

```
2023-10-18 17:05:23 Unsupported cipher in --data-ciphers: Kyber512
Options error: NCP cipher list contains unsupported ciphers or is too long.
Use --help for more information.
```

Figure 1: OpenVPN fails to support Kyber

As a result, I switched from the Kyber algorithm (KEMs) to Dilithium algorithm (Signature schemes), which is also from CRYSTALS (Cryptographic Suite for Algebraic Lattices), the same team that developed Kyber. The reason of it is because I think it is possible to use the generated certificates and keys by a post quantum signature schemes algorithm to secure the VPN application. Therefore, I created certificates and keys for the CA (Certificate Authority), the server, and the client using OpenSSL with the OQS-Provider that provides me with the Dilithium3 algorithm. However, the OpenVPN protocol does not accept the certificates that are generated as it cannot recognize the type of certificates. Although it might be possible to bypass the check of OpenSSL, it fails the purpose of the project to provide quantum-resistance security to the current OpenVPN application.

```
OpenSSL: error:0A0000F7:SSL routines::unknown certificate type
Cannot load certificate file server_cert.crt
Exiting due to fatal error
```

Figure 2: OpenSSL Fails to Determine Certificate Type

After a discussion with the supervisor, I got a suggestion to test out if the certificates generated by the Dilithium3 algorithm are accepted by OpenSSL itself, instead of the OpenVPN protocol. The reason for this is because OpenVPN relies on OpenSSL for their encryption algorithms. Still, I ran into the problem that OpenSSL is unable to start a client server connection with certificates and keys generated by itself. Lastly, after careful investigation and research on OpenSSL and the OQS-Provider library, the problem is solved by using the latest version of OpenSSL (3.2.0-alpha3-dev) instead of the older version (3.0.9). Additionally, OQS-provider allows OpenSSL to apply a KEM algorithm for connections and verify digital signatures that are created by a Signature Scheme algorithm at the same time. As a result, I created a client-server connection using the Kyber algorithm for key exchange and the Dilithium algorithm for digital signatures by using the OQS-enabled OpenSSL.

The final step is the integration of the post-quantum algorithms with the OpenVPN protocol. Instead of using the previously built OpenVPN, it is required to rebuilt OpenVPN using the current OQS-enabled OpenSSL. After building OpenVPN (OpenVPN 2.7_git) and ensuring that it uses the OQS-enabled OpenSSL, the server successfully recognizes the certificates generated by the Dilithium3 algorithm and listens for connection from the client. On the client side, it is also required for the client device to set up OpenVPN that uses the OQS-enabled OpenSSL before connecting to the server. Finally, **the OpenVPN is built using both the Kyber and Dilithium algorithms, and the client can connect to the server with appropriate certificates and keys while hiding its geolocation.**

udp.port==1194												
No.	Time	Source	Destination	Protocol	Length	Info						
47	2.038716240	96.49.215.150	192.168.0.41	OpenVPN	62	MessageType: P_CONTROL_HARD_RESET_CLIENT_V2						
48	2.039050988	192.168.0.41	96.49.215.150	OpenVPN	70	MessageType: P_CONTROL_HARD_RESET_SERVER_V2						
49	2.045075972	96.49.215.150	192.168.0.41	TLSv1.3	1266							
50	2.045076509	96.49.215.150	192.168.0.41	TLSv1.3	329	Continuation Data						
51	2.045949984	192.168.0.41	96.49.215.150	OpenVPN	66	MessageType: P_ACK_V1						
52	2.047316014	192.168.0.41	96.49.215.150	TLSv1.3	1266	Server Hello, Change Cipher Spec						
53	2.047360859	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data						
54	2.047449805	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data						
55	2.047454752	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data						
56	2.047455289	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data						
57	2.047455931	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data						
58	2.051090585	96.49.215.150	192.168.0.41	OpenVPN	70	MessageType: P_ACK_V1						
59	2.051280767	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data						
60	2.052265603	96.49.215.150	192.168.0.41	OpenVPN	74	MessageType: P_ACK_V1						
61	2.052265772	96.49.215.150	192.168.0.41	OpenVPN	78	MessageType: P_ACK_V1						
62	2.052265795	96.49.215.150	192.168.0.41	OpenVPN	78	MessageType: P_ACK_V1						
63	2.052265817	96.49.215.150	192.168.0.41	OpenVPN	78	MessageType: P_ACK_V1						
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)												
Compression Method: null (0)												
Extensions Length: 1102												
▼ Extension: supported_versions (len=2)	Type: supported_versions (43)											
	Length: 2											
Supported Version: TLS 1.3 (0x0304)												
▼ Extension: key_share (len=1092)	Type: key_share (51)											
	Length: 1092											
▼ Key Share extension	Key Share extension											
	▼ Key Share Entry: Group: kyber768, Key Exchange length: 1088											
Group: kyber768 (572)												
Key Exchange Length: 1088												
Key Exchange: 91f1df2665bb40f1864260bdf8288012a295019b821fdafb870f631636e11a565b5ad96b..												
▼ TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec	Content Type: Change Cipher Spec (20)											
	Version: TLS 1.2 (0x0303)											

Figure 3: Wireshark Packet Capture Showing VPN using Kyber768 for key exchange.

```
Validating certificate extended key usage
++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
VERIFY EKU OK
VERIFY OK: depth=0, CN=SERVERFQDN
Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, peer certificate: 192 bits dilithium3, signature: dilithium3
```

Figure 4: Client showing Dilithium3 certificate type when connected to server.

In brief, I underestimated the overall complexity of the project, especially this milestone. I spent more time than estimated for almost every section, which includes understanding of post-quantum cryptography, setting up the libraries from the OQS project, enabling OQS for OpenSSL, and building OpenVPN with specific version of OpenSSL. After overcoming all the obstacles, I eventually created an OpenVPN application that uses both the Kyber and Dilithium algorithms to ensure it is quantum resistant. This milestone reinforced my understanding of post-quantum cryptography and its practical applications, even in the face of compatibility challenges.

2.5.2.3. Milestone 3: GUI Integration

In Milestone 3, my focus shifted to GUI Integration, where I aimed to create a graphical user interface (GUI) for my application using the PyQt library. While this was my first experience with PyQt, I was pleasantly surprised by how quickly I could develop a basic GUI for the application. The initial steps were relatively smooth. I successfully crafted a simple GUI that aligned with the application's requirements. However, the most time-consuming part of this milestone was translating the wireframes I had designed into the actual GUI. This process involved meticulously implementing the visual elements and functionalities outlined in the wireframes.

Despite my best efforts, the constraints of time became a challenge. I couldn't bring every detail from the wireframes into the final GUI. Nonetheless, I managed to ensure that the core functions were fully functional, and users could interact with the application effectively. Another essential aspect of this milestone was dedicating time to error handling. I wanted to make sure that the program didn't crash unexpectedly. By implementing error-handling mechanisms, I could enhance the overall stability and reliability of the application.

In the end, this milestone was a significant achievement. I successfully created two distinct applications, one for the Client and one for the Server, each equipped with a simple yet functional GUI built with PyQt. Despite the time constraints and the challenges of translating wireframes into a working interface, I emerged with two applications that were ready for use, enhancing the user experience and the overall functionality of the VPN system.

2.5.3. Installation Manuals

The installation guide requires the following requirements:

Requirement	Version
Linux System	Recommended: Fedora 37
Python	3.11.0 or higher
OpenSSL	3.2.0 or higher
OpenVPN	2.X built with OQS-enabled OpenSSL
TLS	v1.3
PyQt	PyQt6
Liqoqs, liboqs-python, OQS-Provider	Latest Version
Additional libraries: libtool, cmake, ninja-build, make, openssl, libssl, pkg-config, libnl3, lzo, pam-devel, git, openssl, openssl-devel, perl, pip	

Note: Most of the requirements stated in this section can be installed by the setup script provided with the project. Please ensure to follow the instructions in [Section 2.5.4.1](#) to run the script to get all the libraries installed and configured. This section is for more details of how each individual library/software can be installed if there are any problems after running the script.

➤ **Linux Machine (Tested on Fedora 37)**

- If you do not have Linux on your local machine, you will need to download the .iso file of your choice of Linux System (Fedora 37 in the project) and install it on your local machine.

For Fedora set up, you can find the .iso file here: <https://fedoraproject.org/spins/kde/>

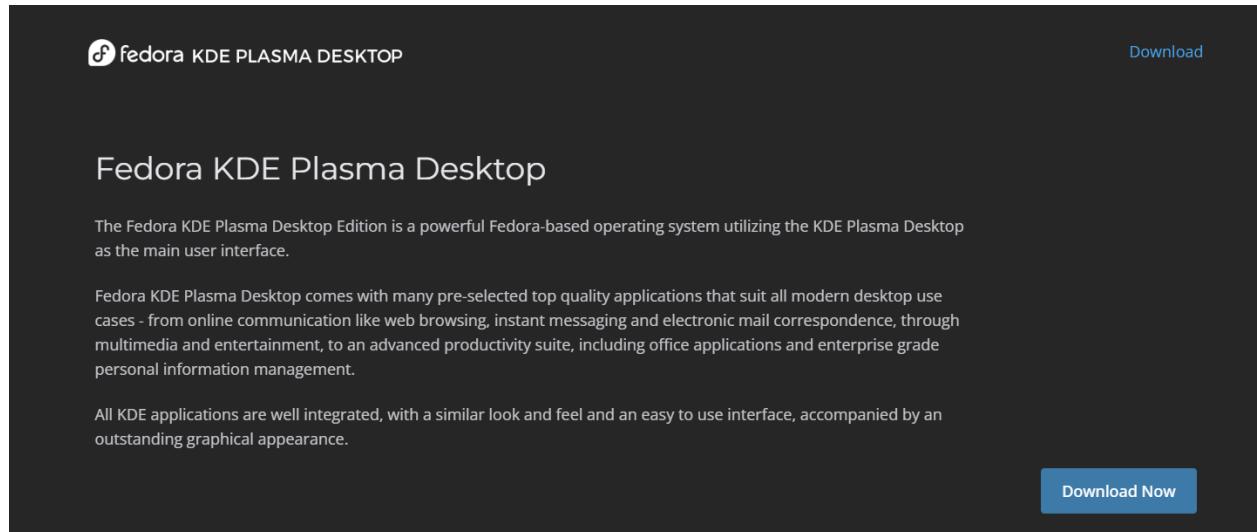


Figure 5: Fedora KDE Plasma Desktop Download Page

- After the .iso file is downloaded, you need to prepare a USB drive and create a bootable image of Linux system to install on your local machine. One of the tools that creates bootable USB drive is Rufus, which can be found in here: <https://rufus.ie/en/>
- After the USB is bootable, restart your device and boot with the USB to install Linux on your local device.

➤ **Python Compiler (Tested on Python 3.11.0)**

- Please check the Python version of your local device before starting the application. You can check the version by entering the command ***python –version***

- **OpenSSL version 3.2.0-alpha3-dev and TLS version 1.3**
 - It is **required** to use the version 3.2.0 or higher as the lower versions are unable to recognize the digital signatures generated by post-quantum algorithm.
- **OpenVPN 2.7_git**
 - It is important to use an OpenVPN 2.X version instead of OpenVPN 3.X as the source code is written in different languages and functions differently.
- **PyQt6**
 - Install the library using the following command: ***pip install PyQt6***
- **OQS-provider**
 - First, clone the Github repository of oqs-provider with the following command:
git clone <https://github.com/open-quantum-safe/oqs-provider.git>
 - Run the setup script and test if it is installed in the repository by the following command:
cd oqs-provider
./scripts/fullbuild.sh
***./scripts/runtests.sh* (Optional)**

Figure 6: Running Tests on oqs-provider after installation.

- Note: By running the ***fullbuild.sh*** script, the program will also install the liboqs library if the device did not have it installed, which is also a required library for the project
 - After we installed OQS-Provider, we need to configure OpenSSL to use the library as a provider.
 - To check if OQS-Provider is installed, we can use the following command:

openssl list -providers

```
sh-5.2# openssl list -providers
Providers:
  default
    name: OpenSSL Default Provider
    version: 3.0.8
    status: active
  oqsprovider
    name: OpenSSL OQS Provider
    version: 0.5.2-dev
    status: active
```

Figure 7: OpenSSL list all the providers

- We can also list out the available signature and KEM algorithms with the following command:

```
openssl list -signature-algorithms -provider oqsprovider
```

```
openssl list -kem-algorithms -provider oqsprovider
```

```
sh-5.2# openssl list -signature-algorithms -provider oqsprovider
{ 1.2.840.113549.1.1.1, 2.5.8.1.1, RSA, rsaEncryption } @ default
{ 1.2.840.10040.4.1, 1.2.840.10040.4.3, 1.3.14.3.2.12, 1.3.14.3.2.1
  .1, 1.3.14.3.2.13, dsaWithSHA, dsaWithSHA1, dsaWithSHA1-old } @ default
{ 1.3.101.112, ED25519 } @ default
{ 1.3.101.113, ED448 } @ default
ECDSA @ default
HMAC @ default
SIPHASH @ default
POLY1305 @ default
CMAC @ default
dilithium2 @ oqsprovider
p256_dilithium2 @ oqsprovider
rsa3072_dilithium2 @ oqsprovider
dilithium3 @ oqsprovider
p384_dilithium3 @ oqsprovider
dilithium5 @ oqsprovider
p521_dilithium5 @ oqsprovider
falcon512 @ oqsprovider
p256_falcon512 @ oqsprovider
rsa3072_falcon512 @ oqsprovider
falcon1024 @ oqsprovider
p521_falcon1024 @ oqsprovider
spincsha2128fsimple @ oqsprovider
p256_spincsha2128fsimple @ oqsprovider
rsa3072_spincsha2128fsimple @ oqsprovider
spincsha2128ssimple @ oqsprovider
p256_spincsha2128ssimple @ oqsprovider
rsa3072_spincsha2128ssimple @ oqsprovider
spincsha2192fsimple @ oqsprovider
p384_spincsha2192fsimple @ oqsprovider
spincshake128fsimple @ oqsprovider
p256_spincshake128fsimple @ oqsprovider
rsa3072_spincshake128fsimple @ oqsprovider
```

Figure 8: OpenSSL list signature algorithms with OQS-provider

```

sh-5.2# openssl list -kem-algorithms -provider oqsprovider
{ 1.2.840.113549.1.1.1, 2.5.8.1.1, RSA, rsaEncryption } @ default
frodo64aes @ oqsprovider
p256_frodo64aes @ oqsprovider
x25519_frodo640aes @ oqsprovider
frodo64shake @ oqsprovider
p256_frodo640shake @ oqsprovider
x25519_frodo640shake @ oqsprovider
frodo976aes @ oqsprovider
p384_frodo976aes @ oqsprovider
x448_frodo976aes @ oqsprovider
frodo976shake @ oqsprovider
p384_frodo976shake @ oqsprovider
x448_frodo976shake @ oqsprovider
frodo1344aes @ oqsprovider
p521_frodo1344aes @ oqsprovider
frodo1344shake @ oqsprovider
p521_frodo1344shake @ oqsprovider
kyber512 @ oqsprovider
p256_kyber512 @ oqsprovider
x25519_kyber512 @ oqsprovider
kyber768 @ oqsprovider
p384_kyber768 @ oqsprovider
x448_kyber768 @ oqsprovider
x25519_kyber768 @ oqsprovider
p256_kyber768 @ oqsprovider
kyber1024 @ oqsprovider
p521_kyber1024 @ oqsprovider
bikel1 @ oqsprovider
p256_bikel1 @ oqsprovider
x25519_bikel1 @ oqsprovider
bikel3 @ oqsprovider
p384_bikel3 @ oqsprovider
x448_bikel3 @ oqsprovider
bikel5 @ oqsprovider
p521_bikel5 @ oqsprovider
hqc128 @ oqsprovider
p256_hqc128 @ oqsprovider
x25519_hqc128 @ oqsprovider
hqc192 @ oqsprovider
p384_hqc192 @ oqsprovider
x448_hqc192 @ oqsprovider
hqc256 @ oqsprovider
p521_hqc256 @ oqsprovider

```

Figure 9: OpenSSL list KEM algorithms with OQS-provider

- In two figures about, it shows that the algorithms from the OQS-provider are available for OpenSSL.
- This can be the end for setting up OQS-provider, but we need to add the -provider flag every time we are executing the commands for OpenSSL. The following instructions show how OpenSSL can use the post-quantum algorithms without having to add the provider flag.
- To enable the provider automatically, we need to go to the configuration file for OpenSSL.

For Fedora 36 users, the configuration file is located in: **/etc/ssl/openssl.cnf**

- In the configuration file, we need to insert a few lines at around line 60-70:

```
oqsprovider = oqsprovider_sect  
[oqsprovider_sect]  
activate = 1  
  
59 [provider_sect]  
60 default = default_sect  
61 ##legacy = legacy_sect  
62 oqsprovider = oqsprovider_sect  
63 ##  
64 [default_sect]  
65 activate = 1  
66 ##  
67 ##[legacy_sect]  
68 ##activate = 1  
69  
70 [oqsprovider_sect]  
71 activate = 1  
72  
73 [ ssl_module ]  
74
```

Figure 10: OpenSSL Configuration for OQS-provider

- After adding those lines, save the configuration file and execute the following command to make sure OpenSSL automatically loads the provider when the command is called:

openssl list -signature-algorithms

openssl list kem-algorithms

```
sh-5.2# openssl list -signature-algorithms
{ 1.2.840.113549.1.1.1, 2.5.8.1.1, RSA, rsaEnc
{ 1.2.840.10040.4.1, 1.2.840.10040.4.3, 1.3.14
tion-old, dsaWithSHA, dsaWithSHA1, dsaWithSHA1-o
{ 1.3.101.112, ED25519 } @ default
{ 1.3.101.113, ED448 } @ default
ECDSA @ default
HMAC @ default
SIPHASH @ default
POLY1305 @ default
CMAC @ default
dilithium2 @ oqsprovider
p256_dilithium2 @ oqsprovider
rsa3072_dilithium2 @ oqsprovider
dilithium3 @ oqsprovider
p384_dilithium3 @ oqsprovider
dilithium5 @ oqsprovider
p521_dilithium5 @ oqsprovider
falcon512 @ oqsprovider
p256_falcon512 @ oqsprovider
rsa3072_falcon512 @ oqsprovider
falcon1024 @ oqsprovider
p521_falcon1024 @ oqsprovider
sphincssha2128fsimple @ oqsprovider
p256_sphincssha2128fsimple @ oqsprovider
rsa3072_sphincssha2128fsimple @ oqsprovider
sphincssha2128ssimple @ oqsprovider
p256_sphincssha2128ssimple @ oqsprovider
rsa3072_sphincssha2128ssimple @ oqsprovider
sphincssha2192fsimple @ oqsprovider
p384_sphincssha2192fsimple @ oqsprovider
sphincsshake128fsimple @ oqsprovider
p256_sphincsshake128fsimple @ oqsprovider
rsa3072_sphincsshake128fsimple @ oqsprovider
```

Figure 11: OpenSSL list signature algorithms without provider

- Finally, Figure 11 shows that OpenSSL automatically loads the OQS-provider when the command is called.

➤ **liboqs-python**

- Note: liboqs-python requires the liboqs library to be installed, which is already been installed when installing oqs-provider above
- First, set the LD_LIBRARY_PATH environmental variable point to the path to liboqs library directory. For Fedora 36 users, the variables can be set with this command:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib64
```

- Second, clone the Github repository of liboqs-python with the following command:


```
git clone --depth=1 https://github.com/open-quantum-safe/liboqs-python
```

- Finally, move into the library directory and install the wrapper with the following command:

```
cd liboqs-python
```

```
pip install .
```

- You can test if the library is configured and installed with the following scripts:

```
python3 liboqs-pytonn/examples/kem.py
```

```
python3 liboqs-pytonn/examples/sig.py
```

```
python3 liboqs-pytonn/examples/rand.py
```

```

sh-5.2# python3 examples/kem.py
/usr/local/lib/python3.10/site-packages/oqs/oqs.py:67: UserWarning: Please install liboqs-python using setup.py
  warnings.warn("Please install liboqs-python using setup.py")
/usr/local/lib/python3.10/site-packages/oqs/oqs.py:74: UserWarning: liboqs version 0.9.0 differs from liboqs-python version None
  warnings.warn("liboqs version {} differs from liboqs-python version {}".format(oqs_version(), oqs_python_version()))
liboqs version: 0.9.0
liboqs-python version: None
Enabled KEM mechanisms:
['BIKE-L1', 'BIKE-L3', 'BIKE-L5', 'Classic-McEliece-348864',
 'Classic-McEliece-348864f', 'Classic-McEliece-460896',
 'Classic-McEliece-460896f', 'Classic-McEliece-6688128',
 'Classic-McEliece-6688128f', 'Classic-McEliece-6960119',
 'Classic-McEliece-6960119f', 'Classic-McEliece-8192128',
 'Classic-McEliece-8192128f', 'HQC-128', 'HQC-192', 'HQC-256', 'Kyber512',
 'Kyber768', 'Kyber1024', 'sntrup761', 'FrodoKEM-640-AES', 'FrodoKEM-640-SHAKE',
 'FrodoKEM-976-AES', 'FrodoKEM-976-SHAKE', 'FrodoKEM-1344-AES',
 'FrodoKEM-1344-SHAKE']

Key encapsulation details:
{'claimed_nist_level': 1,
 'is_ind_cca': True,
 'length_ciphertext': 768,
 'length_public_key': 800,
 'length_secret_key': 1632,
 'length_shared_secret': 32,
 'name': 'Kyber512',
 'version': 'https://github.com/pq-crystals/kyber/commit/74cad307858b61e434490c75f812cb9b9ef7279b'}
b'\x81\x94\x15\xafH0\xd9\x03!VG\xa0\x9c\xadI\x80\xc9\xd2:\x19\xbd\x04,\x82A\x8c\xb1\x840\xfd$\x95\xe5\x11:\x93\x11\x1e\xd6\x12\x
'xa8\xp0\x85\xcc4\x08V\x6dgX\x14\x8cIxY\xc9I\xdcI\x93\xf8G\xb2\x87\x8a\xc7\x02CBz\x99,[\x93\xd0\x990\xc2\xab\xc6{H\x91\x89
'xa8#\x9c\x2\x83l\xfb\xb1G\x98\xbbL< f3\xb1\x04\x072W\x82la\x9a\xf5\x98<\x80VAH?\x8e\xb75\x82\x9c\xc5\x04\xf8\x91~\x1c6\xcf\x95\x
'd%\xac\xb3\xf7\xad\x7j\x84' 3\x5\x95\x00}\xf4C\xaeD+\xe8\x04o\xd4\x1a\xb9V\x93\x8c\xf8\xd4\xbb6\x8b~\xd7\xdak0\xd9Y^'\xfbN\xbe\x
5\xe6\xb5\xd2x\x9b\xd5\x1a\x02\xdfeD\x95\x10P\x96(s9\xe5\x98\xe2\x97h\xba\x0b\x8bY\xb1\x84\xea\x08\xaf\xb3 \x0eW\x4*,\x97\x89\
\xd6\x19\x81\xfc6\x19\x91\x13i\x81\x7\x9c\x90\x99\x00\x82\x0e\x8dy\xb2\xe3h\x9dK<\xab\xf9"\x3\x16P\xe0\xb5\x7f\xf5\x9b|\x
'xb3b\xd9E\xc2\x0c\xf3\xcd\x12\xb7>V\xd9\x11KW\x92\x17\x88\xb4%r\x88hD\x92\xd93\xb9\xae\xcb\x86,\xf4s2\xd4 I;s\xbes-\x13\x99\x9a\x
'C\x16\x06q\x98<\xid\xc5K\x18e\xf2\x0f\x7f\x9aT\x00\xf9\xc4\xbb\x97H\x92\xc2\xc2y'| \x97\x12\x81W\x8bG\x93\x88\x912;= \x7\xd8|\xa
'f81(\xe5\x14\x90T\xbf\x7f\xe0\xc1/\x9a\xb\xt\xnT\xe7\xba\x9ffTC\x10\xae\x87@U\x03\xfa\xc4\xbd\x00\x14k\x85\x95XG\xab\xb2\x8e\x87\xc
'c5\x0ecP\xc7\x18d\x05K\xfa1\x05p\x14\xb7\xe1\xb2\xe1\xf3\x0b\x9a\x0s8g\xb0[(\x10\x96\x9a]\x93\xb0\x89\x16\x95\x90\xaf\xe5\x12\
'7\xf4\xe0\x5g\xbd\xaf\xc1\x94\x060\x96\x90r'
Shared secretes coincide: True

```

Figure 12: Sample script output on liboqs library after installation.

2.5.4. Setup Guide

2.5.4.1. Setup Script

- The setup script enables the user to install and configure all requirements for the VPN application **for both the client and the server.**
- The script is divided into several parts:
 - Installing necessary libraries
 - Building liboqs (Client/Server)
 - Building OpenSSL (Client/Server)
 - Building OQS-Provider (Client/Server)
 - Enabling OQS-OpenSSL (Client/Server)
 - Building OpenVPN based on enabled OQS-OpenSSL (Client/Server)
 - Applying iptables rules for the server to forward client requests (Server)
 - Generating Keys and Certificates with the Dilithium3 algorithm (Server)
- The script can be executed by the following command:
 - Client: ***./client_setup.sh***
 - Server: ***./server_setup.sh***

2.5.4.2. Port Forwarding on Home Router (Server only)

- If you wish to let your client connect to your server from the outside world, you will need to setup port forwarding on your home router on your server's device to allow outside connections.
- To start with, we need the server's device connects to the router (either through LAN or Wi-Fi)
- Afterward, you need to access the router through the gateway. You can find the gateway by the following command: *ip r*

```
sh-5.2# ip r
default via 192.168.0.1 dev enp0s3 proto dhcp src 192.168.0.37 metric 100
```

Figure 13: Finding Network Gateway

- Go to your browser and type the gateway in. In the demonstration, the gateway is 192.168.0.1. Routers might have different login page than the demonstration.

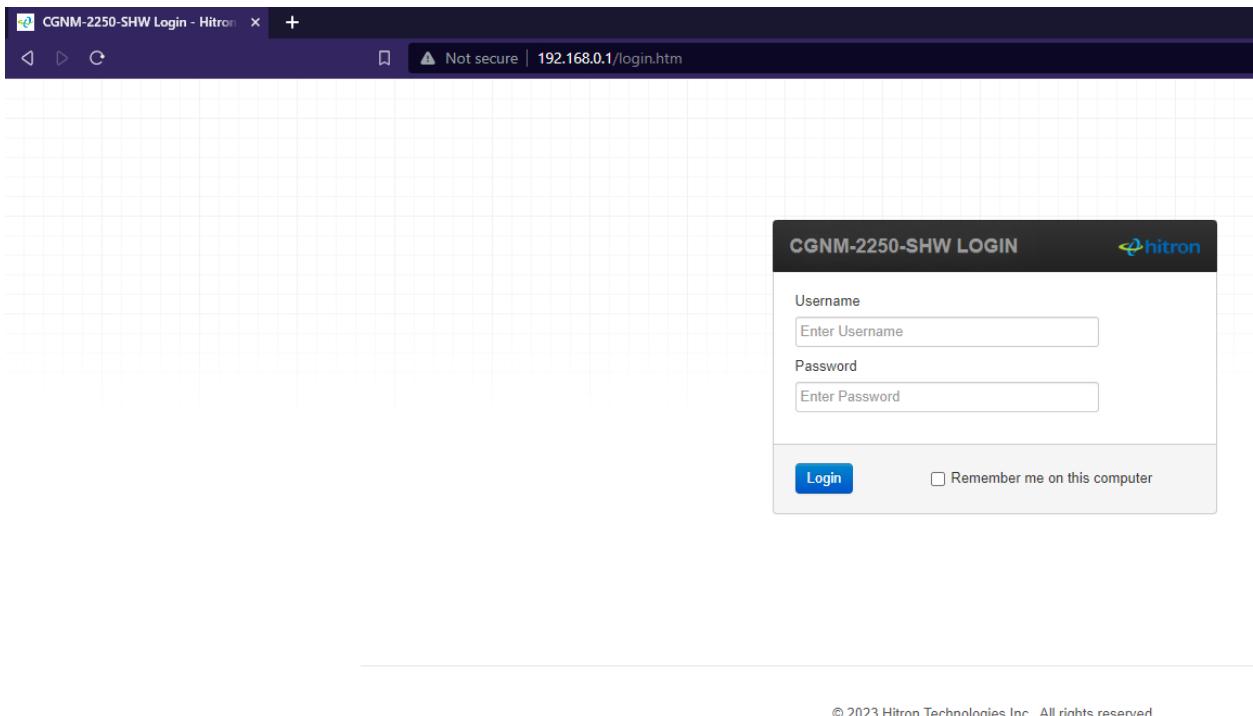


Figure 14: Entering Router IP address / Gateway

- Login with the credentials for the router. If you do not know the credentials, it is often written on the home router.

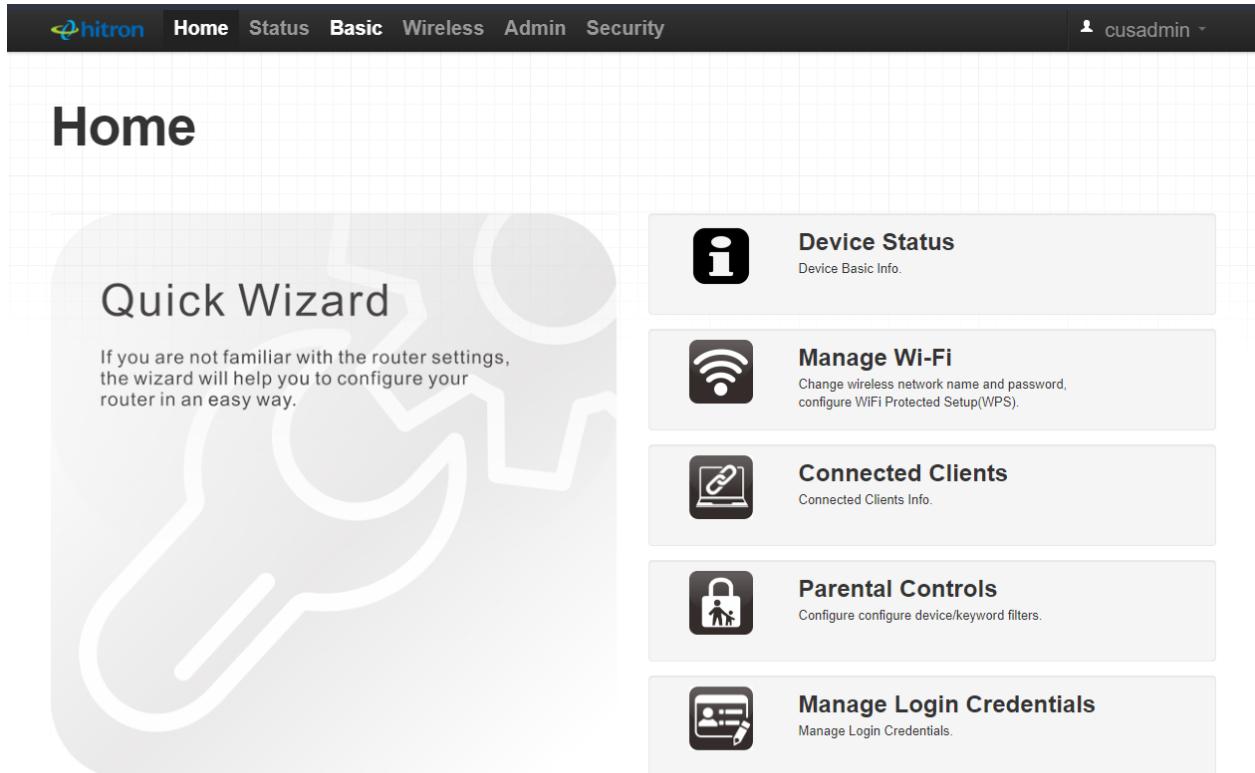


Figure 15: Log Into Router Settings

- Once you are logged in, find the option **Port Forwarding**. In the demonstration, it is under **Basic > Port Forwarding**.

Figure 16: Finding the Port Forwarding Option

- Depending on the router, find the place to add the port forwarding rule for your server.

Add a rule for port forwarding services by user

Common Application	-SERVICES-	
Application Name	OpenVPN	
Protocol	UDP	
Public Port Range	1194	~ 1194
Private Port Range	1194	~ 1194
Local IP Address	192.168.0.31	
Remote IP Address	Any	Specific
Rule Status	ON	OFF
<input type="button" value="Apply"/> <input type="button" value="Close"/>		

Figure 17: Adding Port Forwarding Rule to Server

- Finally, save the changes to the rules and restart the router.
- After the port forwarding rule is applied, your VPN server will be able to accept client connections from the outside world.

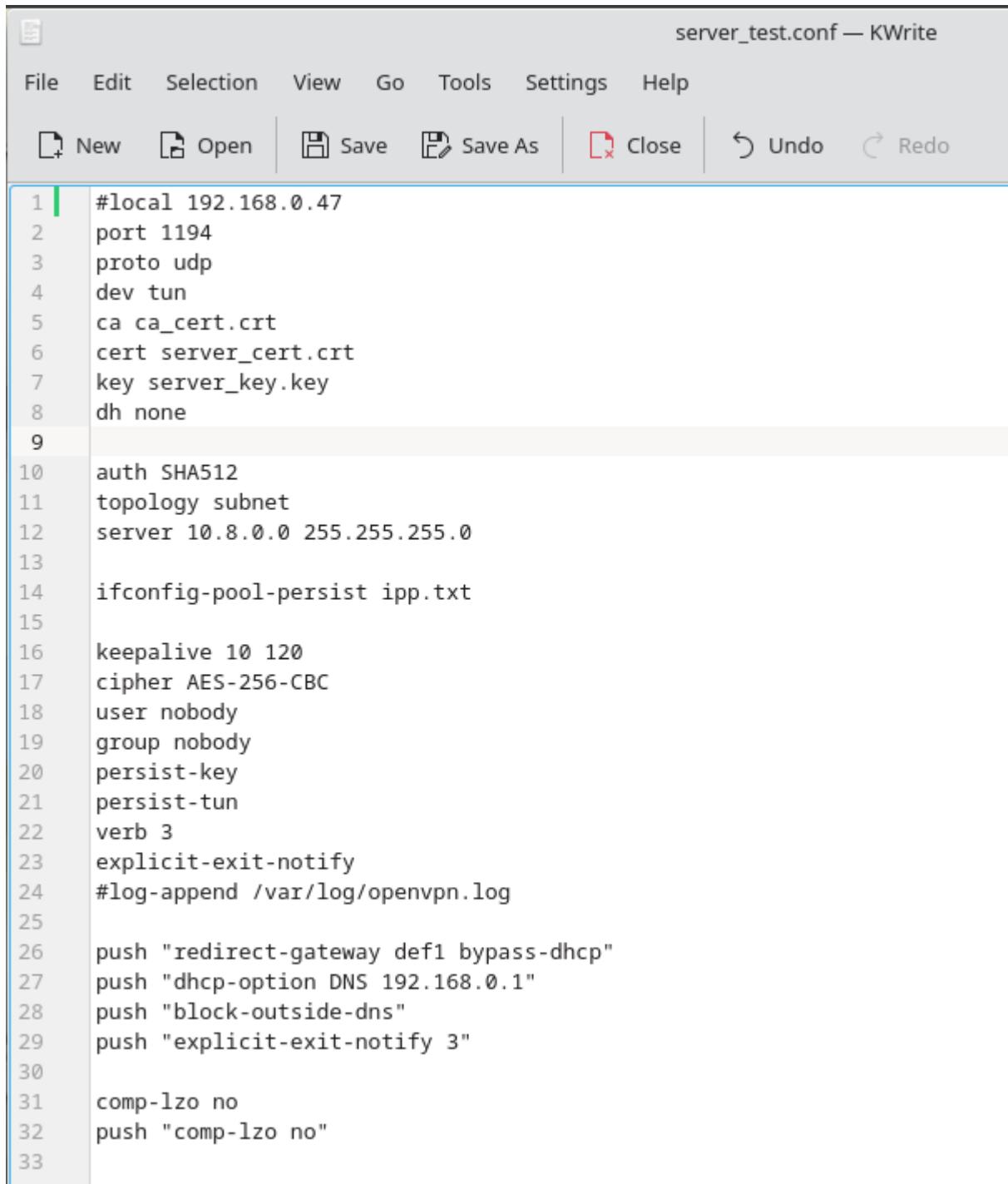
2.5.5. User Manuals

2.5.5.1. VPN application (OpenVPN Protocol)

Server

- If you follow the Setup Guide to install the application in [Section 2.5.4.1.](#), you should have your server installed and configured.
- To start the server application, it is **required** that all the following files are in the **same** directory:
 - openvpn_server.py
 - server.conf
 - ca_cert.crt
 - server_cert.crt
 - server_key.key

- If one of the files is missing, the VPN application will not start. If you wish to change the location of those files, you can go inside the server configuration file and edit the location of each file.



```

server_test.conf — KWrite

File Edit Selection View Go Tools Settings Help
New Open Save Save As Close Undo Redo

1 #local 192.168.0.47
2 port 1194
3 proto udp
4 dev tun
5 ca ca_cert.crt
6 cert server_cert.crt
7 key server_key.key
8 dh none
9
10 auth SHA512
11 topology subnet
12 server 10.8.0.0 255.255.255.0
13
14 ifconfig-pool-persist ipp.txt
15
16 keepalive 10 120
17 cipher AES-256-CBC
18 user nobody
19 group nobody
20 persist-key
21 persist-tun
22 verb 3
23 explicit-exit-notify
24 #log-append /var/log/openvpn.log
25
26 push "redirect-gateway def1 bypass-dhcp"
27 push "dhcp-option DNS 192.168.0.1"
28 push "block-outside-dns"
29 push "explicit-exit-notify 3"
30
31 comp-lzo no
32 push "comp-lzo no"
33

```

Figure 18: VPN Server Configuration File

- Once the configuration is done, you may start the server program with the following command:

python openvpn_server.py

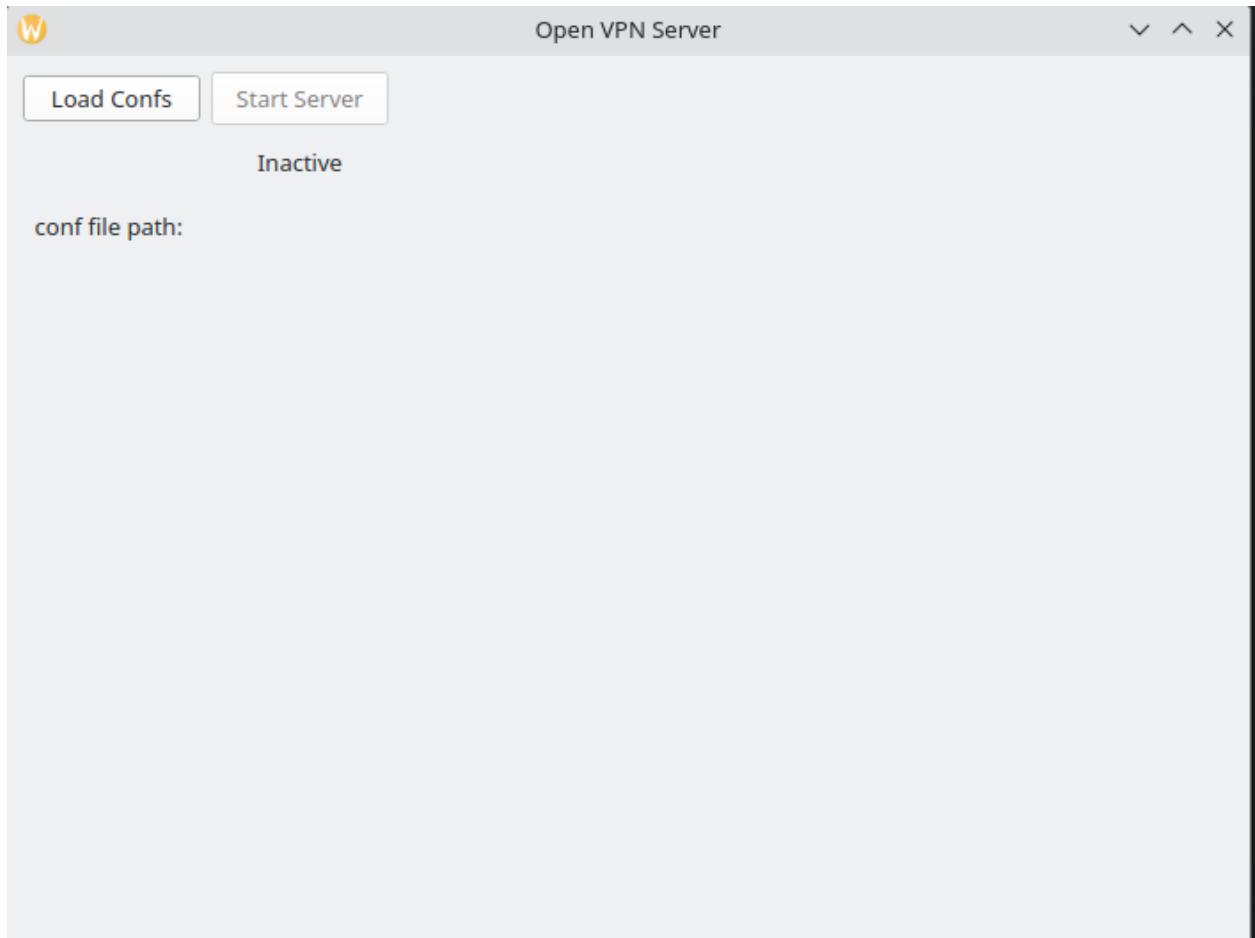


Figure 19: VPN Server GUI

- The GUI of the program will show up after executing the program. Click the “**Load Confs**” button on the **top left** corner of the program and select the configuration file for your OpenVPN Server.

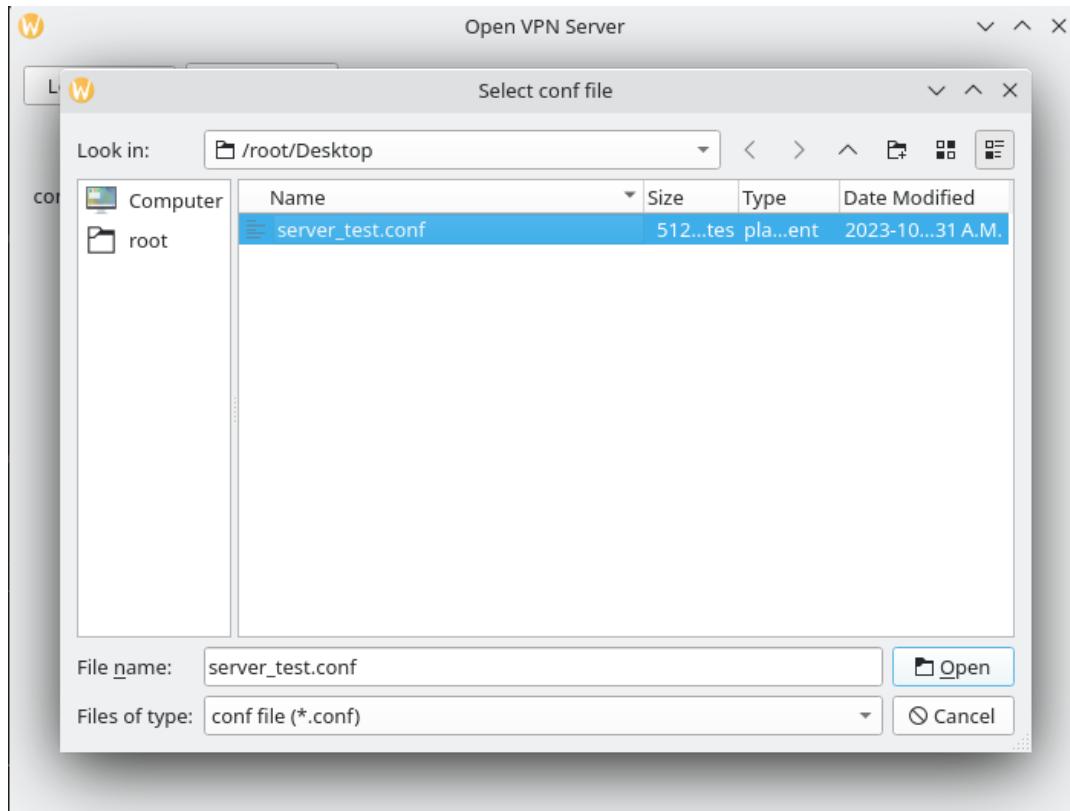


Figure 20: VPN Server- Select Configuration File

- Afterward, the GUI will show the configuration file path. And it is now ready to start the server.

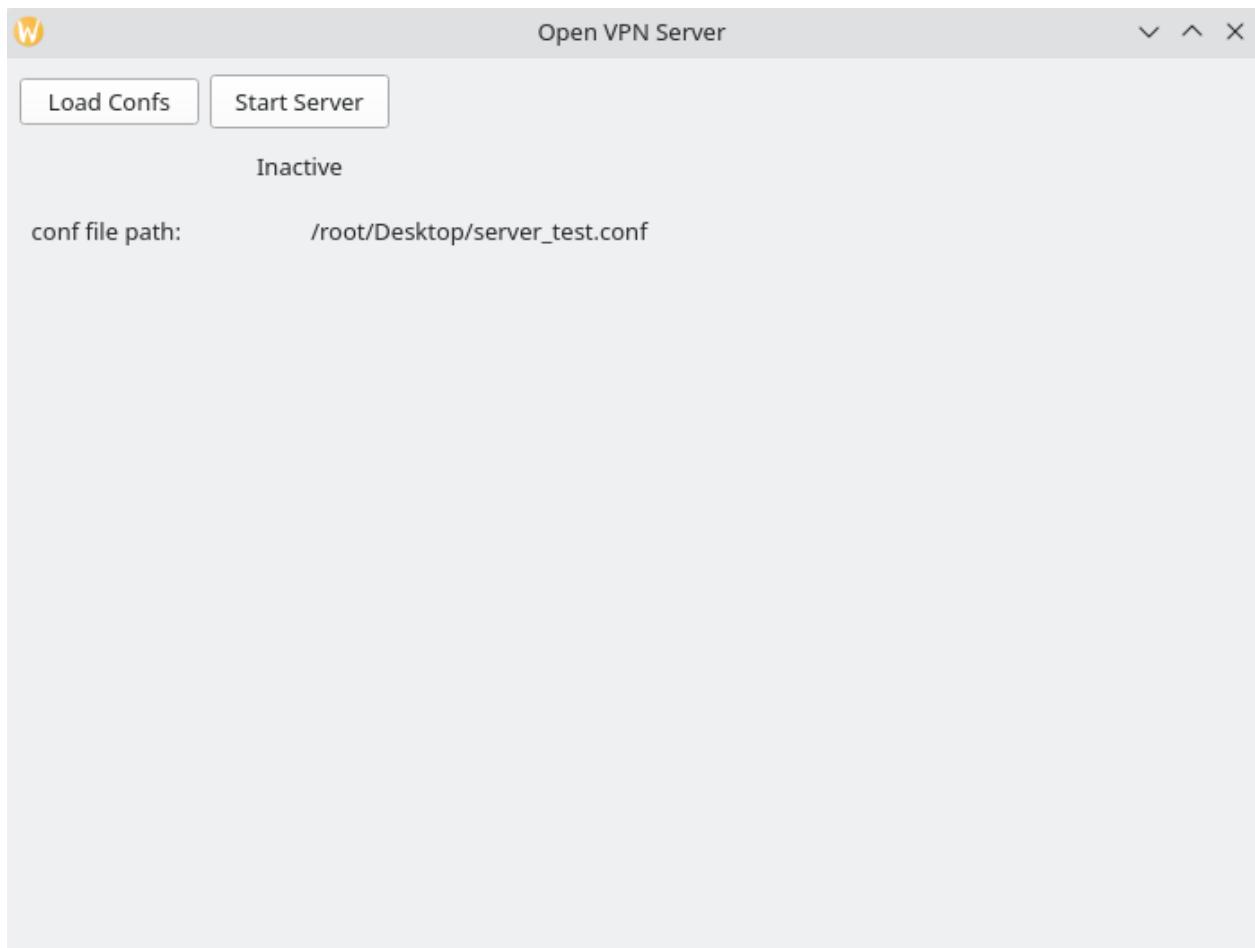
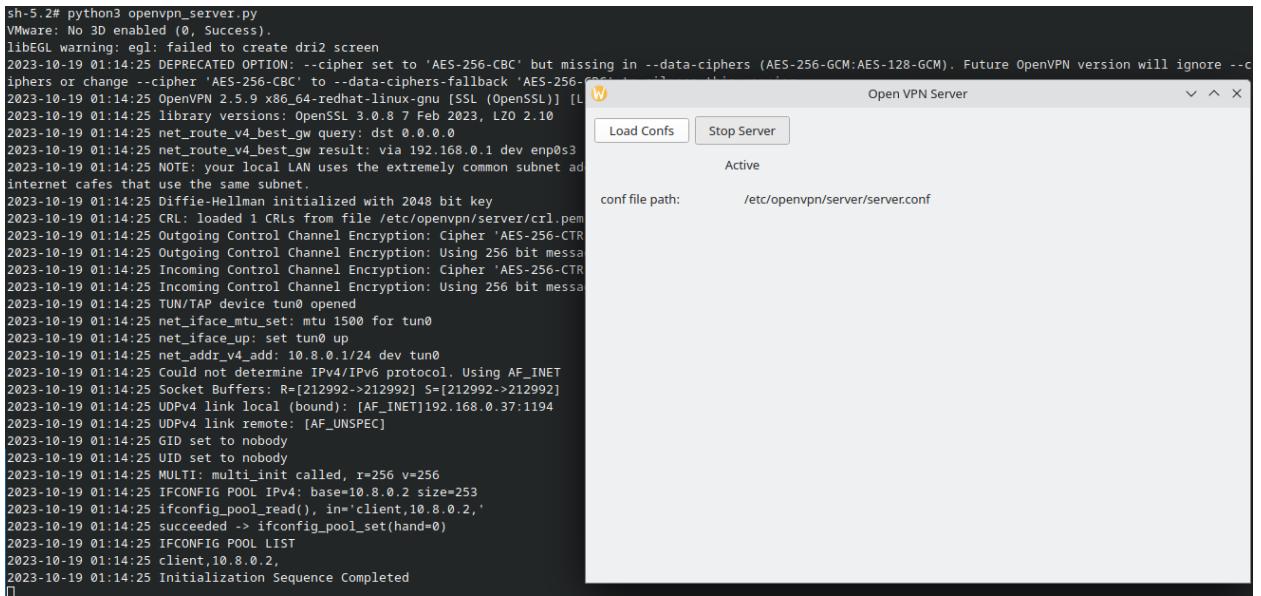


Figure 21: VPN Server- Ready to Start

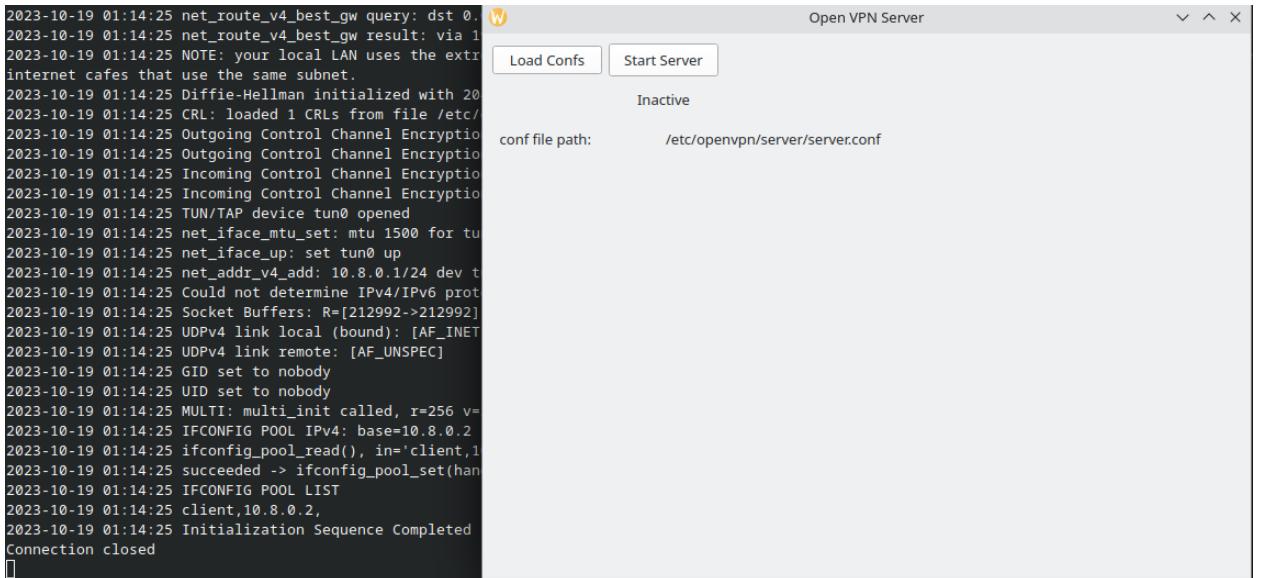
- Finally, click the “Start Server” button to start your own OpenVPN Server!



```
sh-5.2# python3 openvpn_server.py
VMWare: No 3D enabled (0, Success).
libEGL warning: egl: failed to create dri2 screen
2023-10-19 01:14:25 DEPRECATED OPTION: --cipher set to 'AES-256-CBC' but missing in --data-ciphers (AES-256-GCM:AES-128-GCM). Future OpenVPN version will ignore --ciphers or change --cipher 'AES-256-CBC' to --data-ciphers-fallback 'AES-256-GCM' (https://www.openvpn.net/assets/doc/manual/4.4.0/ovpn-manual.html#data-ciphers-fallback)
2023-10-19 01:14:25 OpenVPN 2.5.9 x86_64-redhat-linux-gnu [SSL (OpenSSL)] [LZO] [PKCS11] [MH] [IPv6]
2023-10-19 01:14:25 library versions: OpenSSL 3.0.8 7 Feb 2023, LZO 2.10
2023-10-19 01:14:25 net_route_v4_best_gw query: dst 0.0.0.0
2023-10-19 01:14:25 net_route_v4_best_gw result: via 192.168.0.1 dev enp0s3
2023-10-19 01:14:25 NOTE: your local LAN uses the extremely common subnet address 192.168.0.0/16. Please consider using a different subnet for your internet cafes that use the same subnet.
2023-10-19 01:14:25 Diffie-Hellman initialized with 2048 bit key
2023-10-19 01:14:25 CRL: loaded 1 CRLs from file /etc/openvpn/server/crl.pem
2023-10-19 01:14:25 Outgoing Control Channel Encryption: Cipher 'AES-256-CTR'
2023-10-19 01:14:25 Outgoing Control Channel Encryption: Using 256 bit message authentication code
2023-10-19 01:14:25 Incoming Control Channel Encryption: Cipher 'AES-256-CTR'
2023-10-19 01:14:25 Incoming Control Channel Encryption: Using 256 bit message authentication code
2023-10-19 01:14:25 TUN/TAP device tun0 opened
2023-10-19 01:14:25 net_iface_mtu_set: mtu 1500 for tun0
2023-10-19 01:14:25 net_iface_up: set tun0 up
2023-10-19 01:14:25 net_addr_v4_add: 10.8.0.1/24 dev tun0
2023-10-19 01:14:25 Could not determine IPv4/IPv6 protocol. Using AF_INET
2023-10-19 01:14:25 Socket Buffers: R=[212992->212992] S=[212992->212992]
2023-10-19 01:14:25 UDPv4 link local (bound): [AF_INET]192.168.0.37:1194
2023-10-19 01:14:25 UDPv4 link remote: [AF_UNSPEC]
2023-10-19 01:14:25 GID set to nobody
2023-10-19 01:14:25 UID set to nobody
2023-10-19 01:14:25 MULTI: multi_init called, r=256 v=256
2023-10-19 01:14:25 IFCONFIG POOL IPv4: base=10.8.0.2 size=253
2023-10-19 01:14:25 ifconfig_pool_read(), in='client',10.8.0.2,
2023-10-19 01:14:25 succeeded -> ifconfig_pool_set(hand=0)
2023-10-19 01:14:25 IFCONFIG POOL LIST
2023-10-19 01:14:25 client,10.8.0.2,
2023-10-19 01:14:25 Initialization Sequence Completed
[]
```

Figure 22: VPN Server- Start Server

- You can also click the stop server button at any time to stop your VPN Server after starting it.



```
2023-10-19 01:14:25 net_route_v4_best_gw query: dst 0.0.0.0
2023-10-19 01:14:25 net_route_v4_best_gw result: via 192.168.0.1
2023-10-19 01:14:25 NOTE: your local LAN uses the extremely common subnet address 192.168.0.0/16. Please consider using a different subnet for your internet cafes that use the same subnet.
2023-10-19 01:14:25 Diffie-Hellman initialized with 2048 bit key
2023-10-19 01:14:25 CRL: loaded 1 CRLs from file /etc/openvpn/server/crl.pem
2023-10-19 01:14:25 Outgoing Control Channel Encryption: Cipher 'AES-256-CTR'
2023-10-19 01:14:25 Incoming Control Channel Encryption: Cipher 'AES-256-CTR'
2023-10-19 01:14:25 TUN/TAP device tun0 opened
2023-10-19 01:14:25 net_iface_mtu_set: mtu 1500 for tun0
2023-10-19 01:14:25 net_iface_up: set tun0 up
2023-10-19 01:14:25 net_addr_v4_add: 10.8.0.1/24 dev tun0
2023-10-19 01:14:25 Could not determine IPv4/IPv6 protocol. Using AF_INET
2023-10-19 01:14:25 Socket Buffers: R=[212992->212992] S=[212992->212992]
2023-10-19 01:14:25 UDPv4 link local (bound): [AF_INET]192.168.0.37:1194
2023-10-19 01:14:25 UDPv4 link remote: [AF_UNSPEC]
2023-10-19 01:14:25 GID set to nobody
2023-10-19 01:14:25 UID set to nobody
2023-10-19 01:14:25 MULTI: multi_init called, r=256 v=256
2023-10-19 01:14:25 IFCONFIG POOL IPv4: base=10.8.0.2 size=253
2023-10-19 01:14:25 ifconfig_pool_read(), in='client',10.8.0.2,
2023-10-19 01:14:25 succeeded -> ifconfig_pool_set(hand=0)
2023-10-19 01:14:25 IFCONFIG POOL LIST
2023-10-19 01:14:25 client,10.8.0.2,
2023-10-19 01:14:25 Initialization Sequence Completed
Connection closed
[]
```

Figure 23: VPN Server- Stop Server

Client

- To start the client application, it is **required** that all the following files are in the **same** directory:
 - openvpn_client.py
 - client.ovpn
 - ca_cert.crt
 - client_cert.crt
 - client_key.key
- Also, if your client device is required to have an installed OpenVPN built with the OQS-enabled OpenSSL, which can be done by the setup script for client.
- Once OpenVPN is installed on the device, the client program is ready to start by typing the following command:

python openvpn_client.py

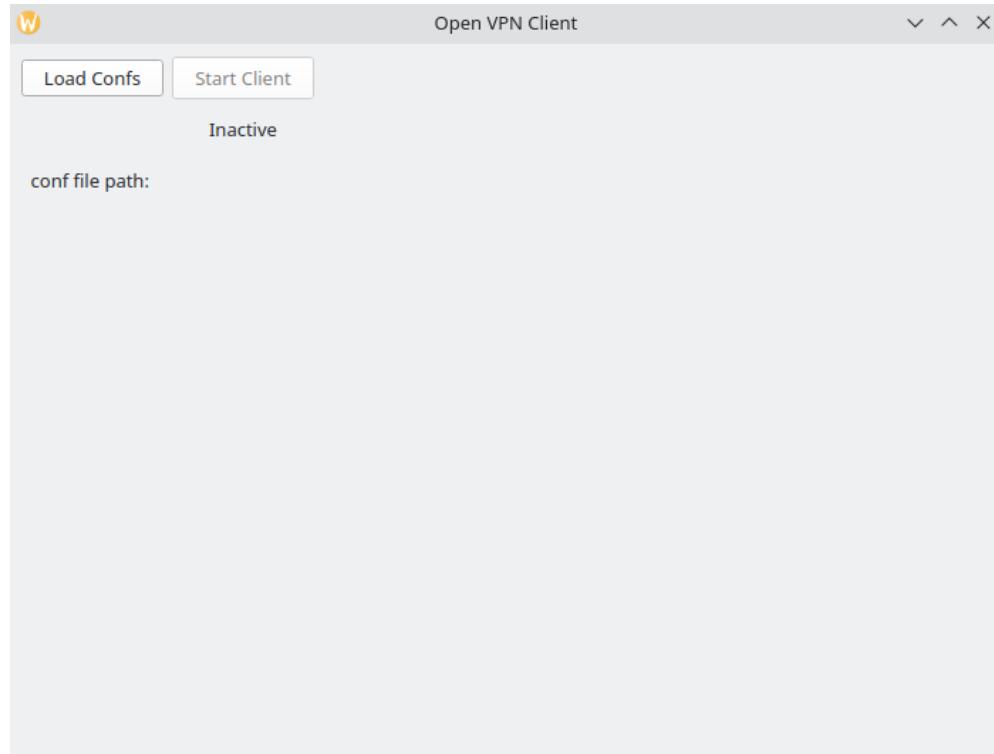


Figure 24: VPN Client GUI

- The GUI of the program will show up after executing the program. Click the “**Load Confs**” button on the **top left** corner of the program and select the configuration file for the OpenVPN client.

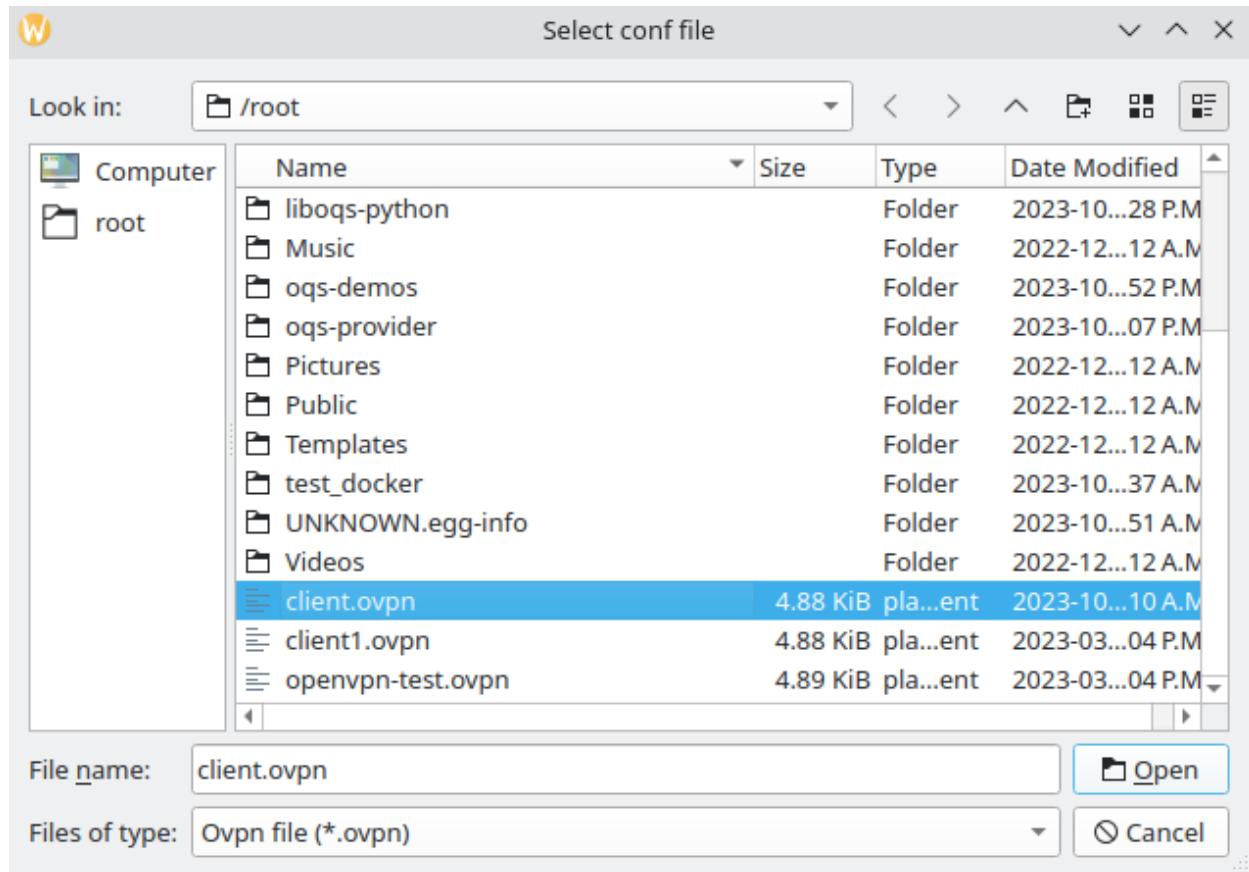


Figure 25: VPN Client- Select Configuration File

- Afterward, the GUI will show the configuration file path. And it is now ready to start the connection.

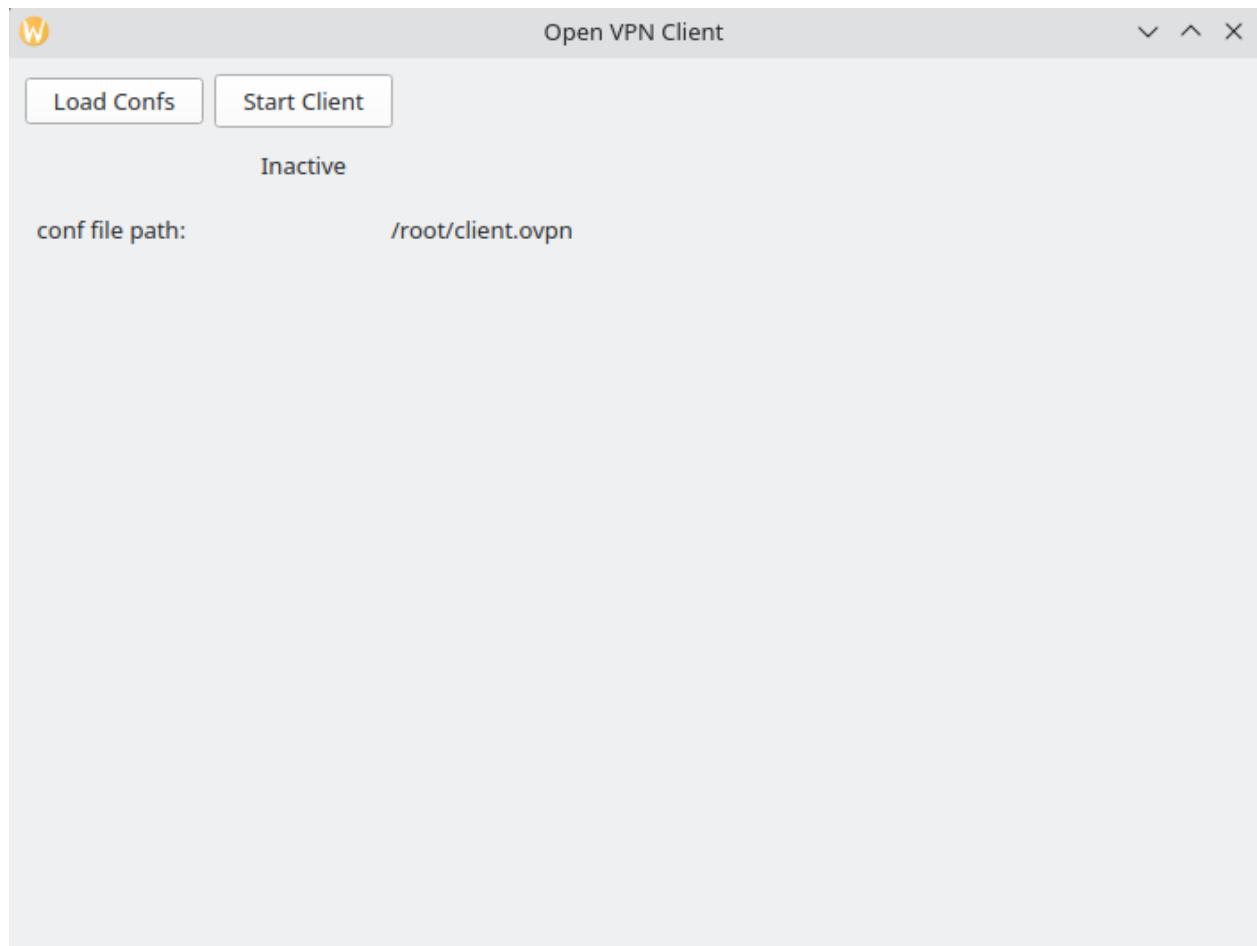


Figure 26: VPN Client- Ready to Start

- Finally, click the “**Start Server**” button to start your own OpenVPN Client!

```

sh-5.2# python3 openvpn_client.py
VMWare: No 3D enabled (0, Success).
libeGGL warning: egl: failed to create dri2 screen
2023-10-19 01:36:01 DEPRECATED OPTION: --cipher set to 'AES-256-CBC' but missing in --dat
iphers or change --cipher 'AES-256-CBC' to --data-ciphers-fallback 'AES-256-CBC' to silent
2023-10-19 01:36:01 OpenVPN 2.5.9 x86_64-redhat-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EP
2023-10-19 01:36:01 library versions: OpenSSL 3.0.8 7 Feb 2023, LZO 2.10
2023-10-19 01:36:01 Outgoing Control Channel Encryption: Cipher 'AES-256-CTR' initialized
2023-10-19 01:36:01 Outgoing Control Channel Encryption: Using 256 bit message hash 'SHA2
2023-10-19 01:36:01 Incoming Control Channel Encryption: Cipher 'AES-256-CTR' initialized
2023-10-19 01:36:01 Incoming Control Channel Encryption: Using 256 bit message hash 'SHA2
2023-10-19 01:36:01 TCP/UDP: Preserving recently used remote address: [AF_INET]96.49.215.
2023-10-19 01:36:01 Socket Buffers: R=[212992->212992] S=[212992->212992]
2023-10-19 01:36:01 UDP link local: (not bound)
2023-10-19 01:36:01 UDP link remote: [AF_INET]96.49.215.150:1194
2023-10-19 01:36:01 TLS: Initial packet from [AF_INET]96.49.215.150:1194, sid=673f6f5b 01
2023-10-19 01:36:01 VERIFY OK: depth=1, CN=Easy-RSA CA
2023-10-19 01:36:01 VERIFY KU OK
2023-10-19 01:36:01 Validating certificate extended key usage
2023-10-19 01:36:01 ++ Certificate has EKU (str) TLS Web Server Authentication, expects T
2023-10-19 01:36:01 VERIFY EKU OK
2023-10-19 01:36:01 VERIFY OK, depth=0, CN=server
2023-10-19 01:36:01 Control Channel: TLSv1.3, cipher TLS_AES_256_GCM_SHA384, peer
2023-10-19 01:36:01 [server] Peer Connection Initiated with [AF_INET]96.49.215.150:1194
2023-10-19 01:36:01 PUSH: Received control message: 'PUSH_REPLY,redirect-gateway def1 bypass
2.255.255.255.0,peer-id 0,cipher AES-256-GCM'
2023-10-19 01:36:01 Unrecognized option or missing or extra parameter(s) in [PUSH-OPTIONS
2023-10-19 01:36:01 OPTIONS IMPORT: timers and/or timeouts modified
2023-10-19 01:36:01 OPTIONS IMPORT: --ifconfig/up options modified
2023-10-19 01:36:01 OPTIONS IMPORT: route options modified
2023-10-19 01:36:01 OPTIONS IMPORT: route-related options modified
2023-10-19 01:36:01 OPTIONS IMPORT: --ip-win32 and/or --dhcp-option options modified

```

Figure 27: VPN Client- Start Client

- In the screenshot above, we can see that the client successfully connects to the server.
- You can also click the “**Stop Client**” button at any time to stop your VPN Client after starting it.

```

2023-10-19 01:36:01 Control Channel: TLSv1.3, cipher TLS_AES_256_GCM_SHA384, peer
2023-10-19 01:36:01 [server] Peer Connection Initiated with [AF_INET]96.49.215.150:1194
2023-10-19 01:36:01 PUSH: Received control message: 'PUSH_REPLY,redirect-gateway def1 bypass
2.255.255.255.0,peer-id 0,cipher AES-256-GCM'
2023-10-19 01:36:01 Unrecognized option or missing or extra parameter(s) in [PUSH-OPTIONS
2023-10-19 01:36:01 OPTIONS IMPORT: timers and/or timeouts modified
2023-10-19 01:36:01 OPTIONS IMPORT: --ifconfig/up options modified
2023-10-19 01:36:01 OPTIONS IMPORT: route options modified
2023-10-19 01:36:01 OPTIONS IMPORT: route-related options modified
2023-10-19 01:36:01 OPTIONS IMPORT: --ip-win32 and/or --dhcp-option options modified
2023-10-19 01:36:01 OPTIONS IMPORT: peer-id set
2023-10-19 01:36:01 OPTIONS IMPORT: adjusting link_mtu
2023-10-19 01:36:01 OPTIONS IMPORT: data channel crypt
2023-10-19 01:36:01 Data Channel: using negotiated cipher
2023-10-19 01:36:01 Outgoing Data Channel: Cipher 'AES
2023-10-19 01:36:01 Incoming Data Channel: Cipher 'AES
2023-10-19 01:36:01 net_route_v4_best_gw query: dst 0.
2023-10-19 01:36:01 net_route_v4_best_gw result: via 1
2023-10-19 01:36:01 ROUTE_GATEWAY 192.168.0.1/255.255.
2023-10-19 01:36:01 TUN/TAP device tun1 opened
2023-10-19 01:36:01 net_iface_mtu_set: mtu 1500 for tun1
2023-10-19 01:36:01 net_iface_up: set tun1 up
2023-10-19 01:36:01 net_addr_v4_add: 10.8.0.2/24 dev tun1
2023-10-19 01:36:01 net_route_v4_add: 96.49.215.150/32
2023-10-19 01:36:01 net_route_v4_add: 0.0.0.0/1 via 10
2023-10-19 01:36:01 net_route_v4_add: 128.0.0.0/1 via 10
2023-10-19 01:36:01 Initialization Sequence Completed
Connection closed

```

Figure 28: VPN Client- Stop Client

2.5.6. System Diagram (VPN Application)

The following diagram is the system diagram for the VPN application with OpenVPN Protocol.

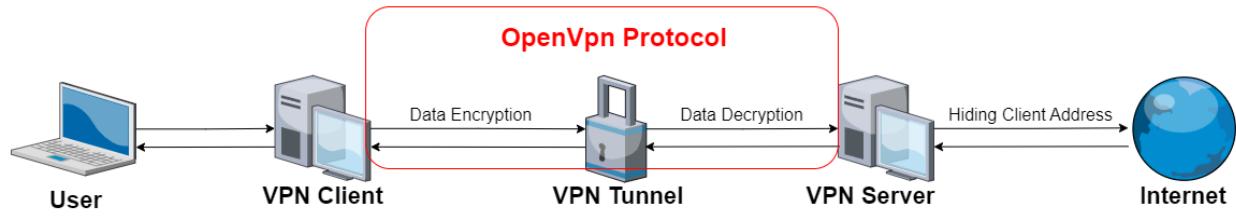


Figure 29: System Diagram of VPN Application

The application is consisting of two parts:

1. **VPN Client:** VPN Client runs on the user's device. It encrypts the data from the user by applying the selected cipher provided by OpenVPN and sends the traffic through VPN Tunnel to the VPN Server.
2. **VPN Server:** VPN Server runs on the server's device. It listens to communications on specified port and receive data. Afterward, it decrypts the received data and makes requests to the internet while hiding the client's address and geolocation.

The application is developed with the OpenVPN protocol and encrypted by the selected cipher provided by OpenSSL. The OpenVPN protocol handles both the certificate signing and verification, as well as the key exchange process during the traffic, to make the entire connection secure.

2.5.7. Software Architecture Diagram (VPN Application)

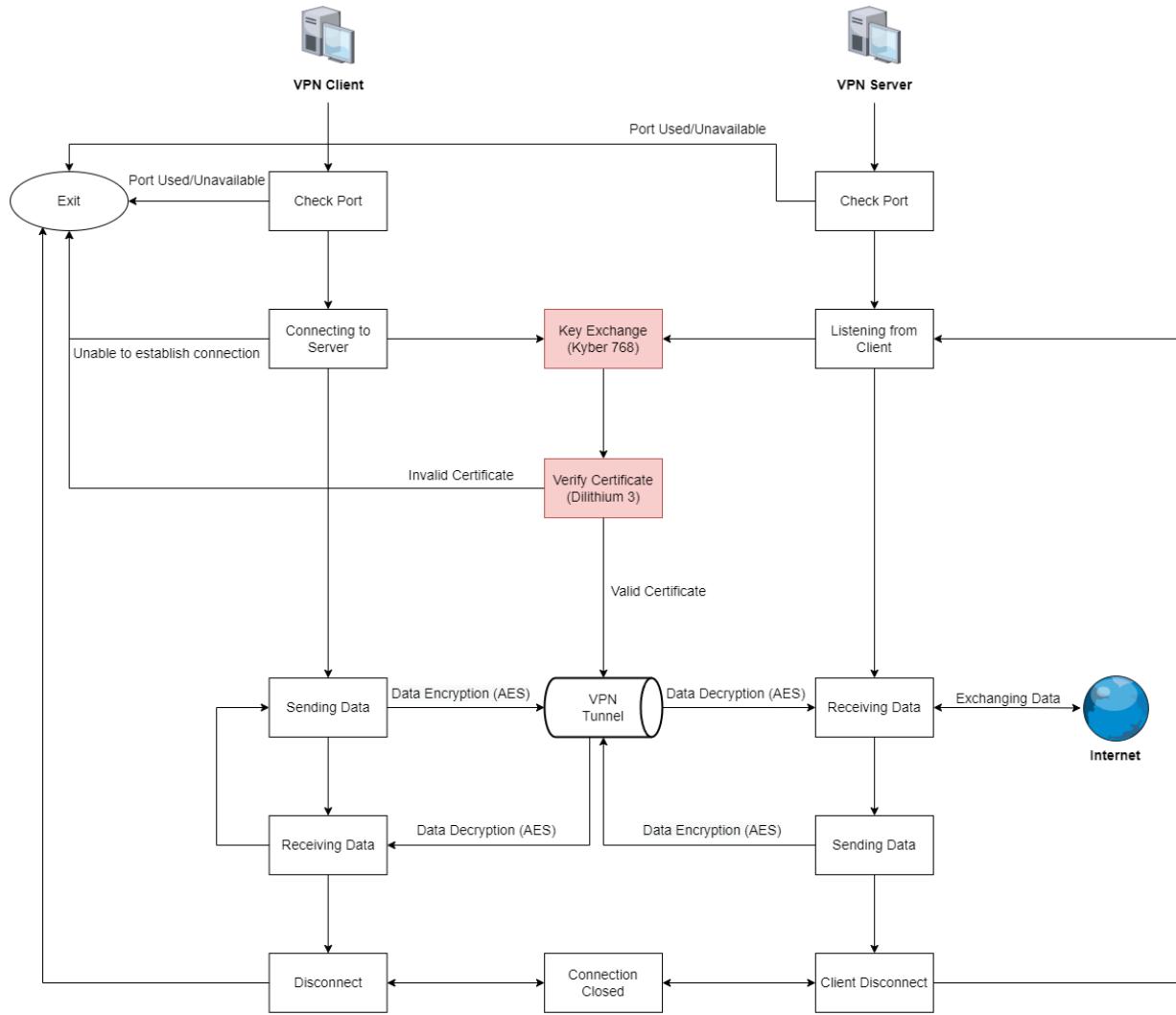


Figure 30: Software Architecture Diagram of VPN Application

Architecture

The architecture of the software is shown in the figure above. The client starts by checking if the port is either used or unavailable. If it is, the program will exit. Otherwise, it continues to the next step to connect to the server. If it is unable to connect to the server on the port, it will exit. The server, at the same time, listens to connections on the same port. If the server receives requests from the client, the server verifies the certificate and responds to the client if it is valid. Once the connection is established,

the client and the server form a VPN tunnel for data transmissions. Data encryption and decryption are done by both ends with their agreement of the encryption ciphers. The server receives the requests from the client and exchange data with the internet. Afterward, the server encrypts the data that it receives from the internet and sends it back to the client. After the client disconnects, the connection is closed.

Kyber and Dilithium

Kyber and Dilithium are post-quantum cryptographic algorithms, specifically designed to resist attacks from quantum computers. Quantum Safe TLS ensures the security of data during its transmission, but it does not provide protection for data when it is at rest. The quantum-safe algorithms incorporated within Key Protect serve as a safeguard for your data, shielding it from potential breaches as it journeys to a Key Protect service endpoint. This protection extends to the encryption of imported root keys and their associated payloads with TLS session keys. These algorithms serve different purposes within the realm of public-key cryptography:

- Kyber (Key Encapsulation Mechanism):

Kyber is a key encapsulation mechanism (KEM) designed to provide secure key exchange. It's used to establish a shared secret key between two parties while protecting the confidentiality and integrity of the key. The core of Kyber is based on lattice-based cryptography, specifically using the Learning With Errors (LWE) problem as a foundation. This problem is believed to be hard for both classical and quantum computers.

Kyber generates public and private key pairs. The public key can be safely shared with others, while the private key is kept secret. When two parties want to establish a shared secret, one party uses the recipient's public key to encapsulate a random secret key, and the recipient uses their private key to decapsulate the shared secret.

- Dilithium (Signature Scheme):

Dilithium is a post-quantum digital signature scheme. Its primary purpose is to provide secure digital signatures, ensuring the authenticity and integrity of data. Like Kyber, Dilithium is also based on lattice-based cryptography, specifically targeting the hardness of the Ring Learning With Errors (Ring-LWE) problem. This problem is resistant to attacks by both classical and quantum computers. Dilithium generates a public and private key pair for digital signatures. To sign a message, the private key holder uses their private key to create a signature for the message. The signature can be verified by anyone using the corresponding public key to ensure the message's authenticity and integrity.

2.5.8. System Use Case Diagram

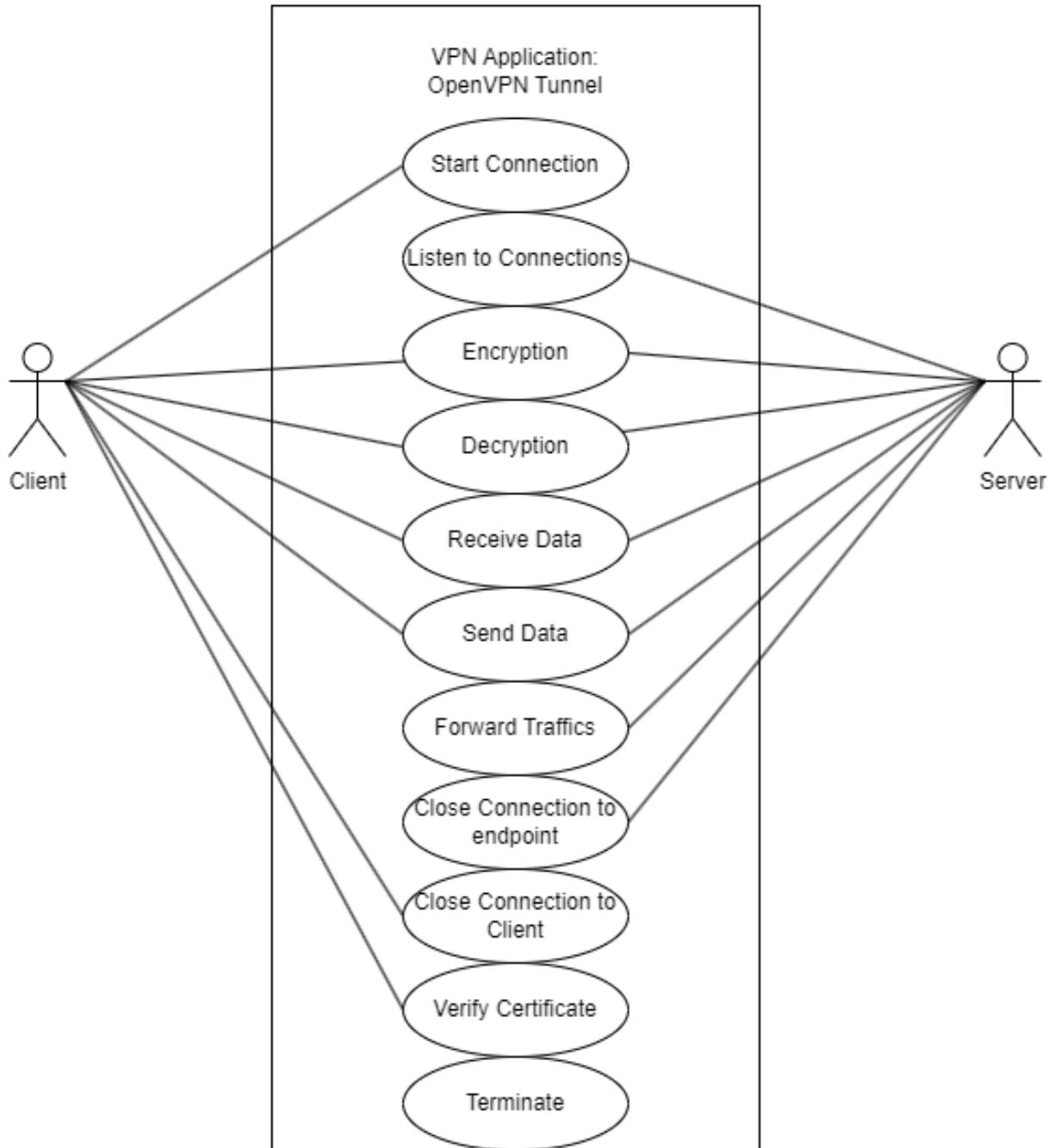


Figure 31: Use Case Diagram of VPN Application

2.5.9. Network Diagram of the VPN Application

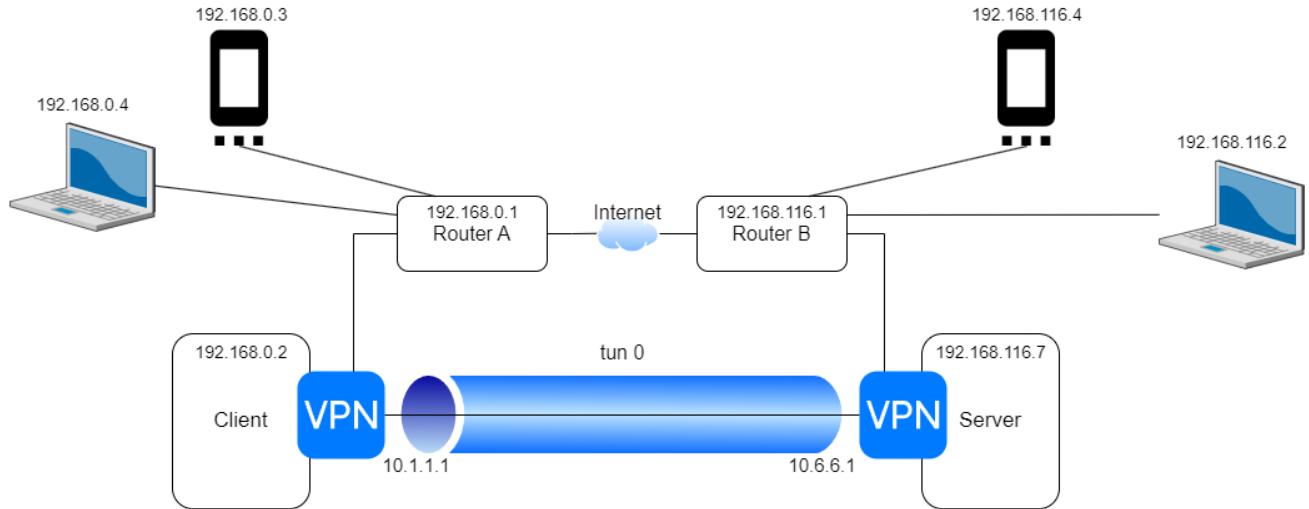


Figure 32: Network Diagram of VPN Application

The Network Diagram shows how the VPN is involved in the entire network.

- Only the Client (192.168.0.2) and the Server (192.168.116.7) have the VPN turned on.
- Both the client and the server require encryption with the AES-256 algorithm in CBC mode. The AES algorithm uses a 256-bit key.
- Both the client and the server require authentication with the SHA-2 algorithm. The SHA-2 algorithm uses a 512-bit key.
- Both the client and the server use shared security associations.

2.5.10. GUI UML Diagram

The figure below shows the UML Diagram for the GUI of the server and the client program. The GUI is built with the PyQt library, which contains several objects for the screen and the elements. It starts with the main window at the bottom that is responsible for the main layout elements such as the buttons and the displayed text. On top of that are the widgets that are responsible for displaying the selection of the config files.

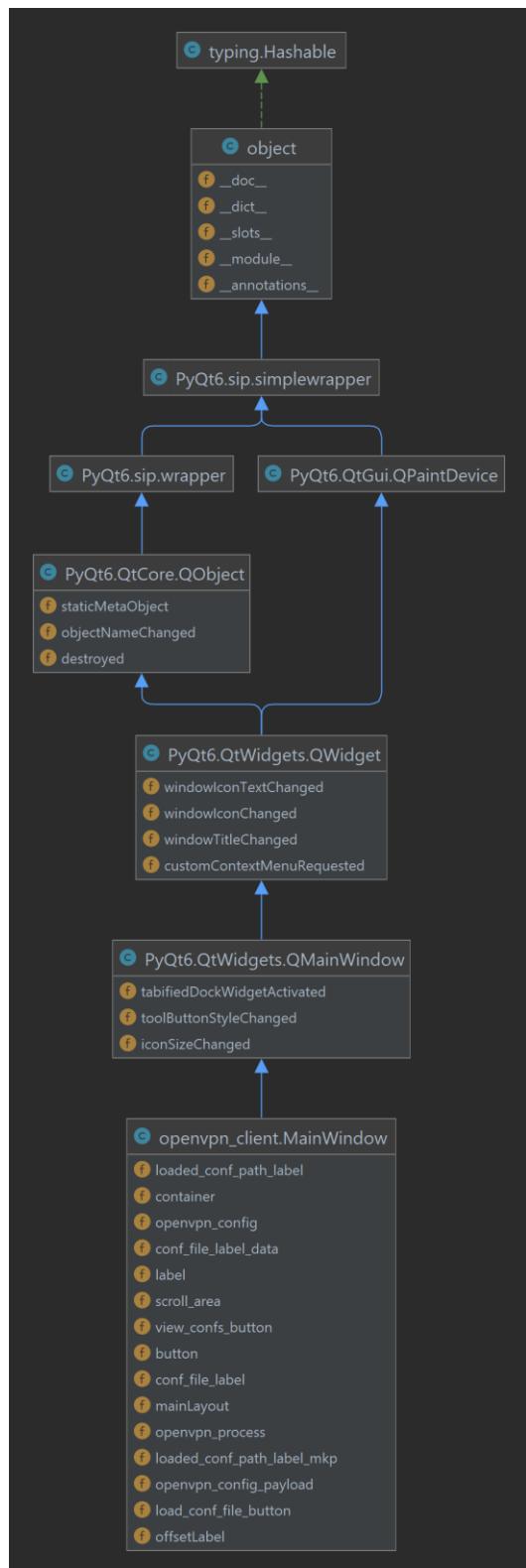


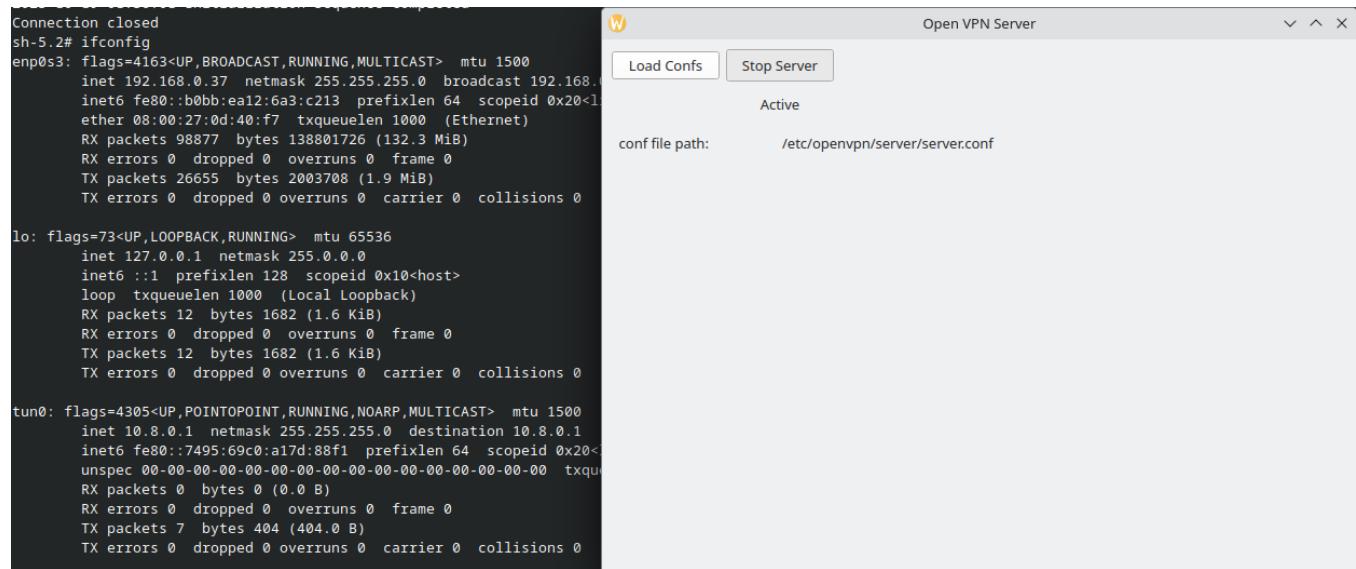
Figure 33: GUI UML Diagram

2.6. Testing Details and Results

2.6.1. VPN with OpenVPN Protocol

Test Case Number	Description	Status
#1	<p>The server's VPN tunneling functions normally.</p> <p>Pass: The tunnel is opened when the server turns on the VPN, and the tunnel is closed when the server turns off.</p> <p>Fail: The tunnel is either closed while server is on or opened while the server is off.</p>	Pass
#2	<p>The client's VPN tunneling functions normally.</p> <p>Pass: The tunnel is opened when the client connects to the server, and the tunnel is closed when the client does not connect to the server.</p> <p>Fail: The tunnel is either closed while client is connected to the server or opened while the client is not connected to the server.</p>	Pass
#3	<p>Traffic is forwarded through the tunnel to the server.</p> <p>Pass: The server receives the network request.</p> <p>Fail: The server does not receive the network request.</p>	Pass
#4	<p>The client's address and geolocation are hidden.</p> <p>Pass: The IP address on the internet is the server's address.</p> <p>Fail: The IP address on the internet is the client's address.</p>	Pass
#5	<p>Two or more clients can secure their addresses and geolocations.</p> <p>Pass: The IP addresses on the internet for all clients are all the server's address.</p> <p>Fail: One or none of the IP addresses are the server's address.</p>	Pass
#6	<p>Data is secured with encryption.</p> <p>Pass: The tokens are all encrypted.</p> <p>Fail: The tokens are not encrypted.</p>	Pass

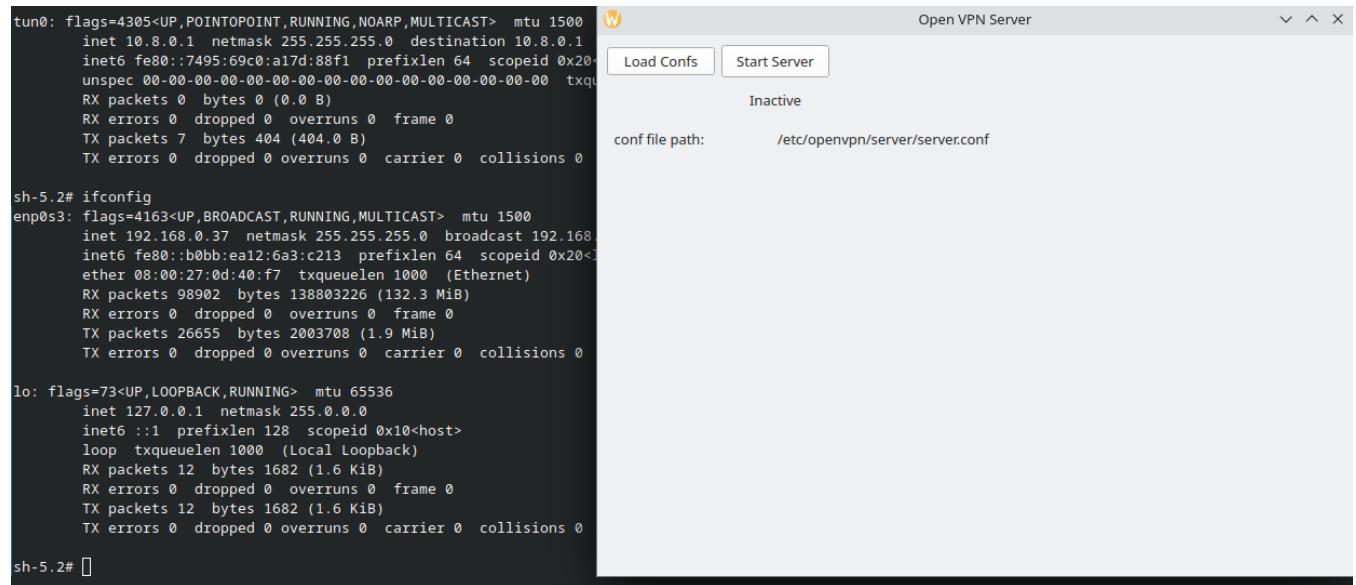
Test Case #1: The server's VPN tunneling functions normally.



Connection closed
sh-5.2# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.37 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::b0bb:ea12:6a3:c213 prefixlen 64 scopeid 0x20<brd
ether 08:00:27:0d:40:f7 txqueuelen 1000 (Ethernet)
RX packets 98877 bytes 138801726 (132.3 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 26655 bytes 2003708 (1.9 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 12 bytes 1682 (1.6 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 12 bytes 1682 (1.6 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
inet 10.8.0.1 netmask 255.255.255.0 destination 10.8.0.1
inet6 fe80::7495:69c0:a17d:88f1 prefixlen 64 scopeid 0x20<brd
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txque
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 7 bytes 404 (404.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0



tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
inet 10.8.0.1 netmask 255.255.255.0 destination 10.8.0.1
inet6 fe80::7495:69c0:a17d:88f1 prefixlen 64 scopeid 0x20<brd
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txque
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 7 bytes 404 (404.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

sh-5.2# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.37 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::b0bb:ea12:6a3:c213 prefixlen 64 scopeid 0x20<brd
ether 08:00:27:0d:40:f7 txqueuelen 1000 (Ethernet)
RX packets 98902 bytes 138803226 (132.3 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 26655 bytes 2003708 (1.9 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 12 bytes 1682 (1.6 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 12 bytes 1682 (1.6 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

sh-5.2# []

The tunnel is created while the server is on, and the tunnel is closed after the server is turned off.

Test Case #2: The client's VPN tunneling functions normally.

The image shows two terminal windows side-by-side. The left terminal window displays the output of the 'ifconfig' command on a Fedora desktop. It lists three interfaces: enp0s3 (ethernet), lo (loopback), and tun0 (VPN tunnel). The tun0 interface is shown with a destination IP of 10.8.0.2 and a netmask of 255.255.255.0. The right terminal window shows the 'Open VPN Client' application window. The 'Load Confs' button is highlighted. The status bar indicates 'Active' and the configuration file path is '/root/Desktop/client_conf/client.ovpn'. The second terminal window shows the same 'ifconfig' output, but the tun0 interface has disappeared, indicating it has been closed after the connection was terminated.

```
[root@fedora Desktop]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.41 brd 192.168.0.255 netmask 255.255.255.0 broadcast 192.168.0.255
        ether fe80::2981:3f7d:a095:de50 brd fe80::ff:fe57:a095
        inet6 fe80::2981:3f7d:a095:de50 brd fe80::ff:fe57:a095 scopeid 0x20<link>
            ether 08:00:27:4b:c3:e0 txqueuelen 1000  (Ethernet)
            RX packets 52279 bytes 78577930 (74.9 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 10760 bytes 746113 (728.6 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
        ether 00:00:00:00:00:00 txqueuelen 1000  (Local Loopback)
        RX packets 58 bytes 5642 (5.5 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 58 bytes 5642 (5.5 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.8.0.2 brd 10.8.0.2 netmask 255.255.255.0 destination 10.8.0.2
        ether fe80::2d0b:e95f:76e9:ac16 txqueuelen 1000  (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 31 bytes 1832 (1.7 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@fedora Desktop]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.41 brd 192.168.0.255 netmask 255.255.255.0 broadcast 192.168.0.255
        ether fe80::2981:3f7d:a095:de50 brd fe80::ff:fe57:a095
            ether 08:00:27:4b:c3:e0 txqueuelen 1000  (Ethernet)
            RX packets 52312 bytes 78580306 (74.9 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 10799 bytes 750097 (732.5 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
        ether 00:00:00:00:00:00 txqueuelen 1000  (Local Loopback)
        RX packets 58 bytes 5642 (5.5 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 58 bytes 5642 (5.5 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Open VPN Client

Load Confs Stop Client

Active

conf file path: /root/Desktop/client_conf/client.ovpn

W

Open VPN Client

Load Confs Start Client

Inactive

conf file path: /root/Desktop/client_conf/client.ovpn

W

The tunnel is opened when the client connects to the server, and the tunnel is closed when the client does not connect to the server.

Test Case #3: Traffic is forwarded through the tunnel to the server.

udp.port == 1194 and udp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	192.168.0.41	96.49.215.150	OpenVPN	98	MessageType: P_CONTROL_HARD_RESET_CLIENT_V2
2	0.0065829590	96.49.215.150	192.168.0.41	OpenVPN	110	MessageType: P_CONTROL_HARD_RESET_SERVER_V2
3	0.0067765320	192.168.0.41	96.49.215.150	OpenVPN	106	MessageType: P_ACK_V1
4	0.0068139350	192.168.0.41	96.49.215.150	SSL	369	Continuation Data
5	0.0152774550	96.49.215.150	192.168.0.41	SSL	1172	Continuation Data
6	0.0152776120	96.49.215.150	192.168.0.41	SSL	1160	Continuation Data
7	0.0155208680	96.49.215.150	192.168.0.41	SSL	357	Continuation Data
8	0.0156074380	192.168.0.41	96.49.215.150	OpenVPN	106	MessageType: P_MESSAGE_V1
9	0.0163116260	192.168.0.41	96.49.215.150	OpenVPN	106	MessageType: P_MESSAGE_V1
10	0.0177541660	192.168.0.41	96.49.215.150	SSL	1172	Continuation Data
11	0.0179187510	192.168.0.41	96.49.215.150	SSL	1160	Continuation Data
12	0.0179250170	192.168.0.41	96.49.215.150	SSL	520	Continuation Data
13	0.0227046530	96.49.215.150	192.168.0.41	OpenVPN	106	MessageType: P_MESSAGE_V1
14	0.0247168770	96.49.215.150	192.168.0.41	SSL	268	Continuation Data
15	0.0248877130	192.168.0.41	96.49.215.150	OpenVPN	106	MessageType: P_MESSAGE_V1
16	0.0249738620	96.49.215.150	192.168.0.41	SSL	331	Continuation Data
17	0.0250789470	192.168.0.41	96.49.215.150	OpenVPN	106	MessageType: P_MESSAGE_V1
18	0.0289802150	96.49.215.150	192.168.0.41	SSL	336	Continuation Data
19	0.0309516270	192.168.0.41	96.49.215.150	OpenVPN	106	MessageType: P_MESSAGE_V1
20	0.0310382160	192.168.0.41	96.49.215.150	OpenVPN	116	MessageType: P_MESSAGE_V1
21	0.0310382160	192.168.0.41	96.49.215.150	OpenVPN	116	MessageType: P_MESSAGE_V1
26	0.0644409866	192.168.0.41	96.49.215.150	OpenVPN	108	MessageType: P_MESSAGE_V1
27	0.064465775	192.168.0.41	96.49.215.150	OpenVPN	144	MessageType: P_MESSAGE_V1
56	0.1403379904	192.168.0.41	96.49.215.150	OpenVPN	128	MessageType: P_MESSAGE_V1
52	0.144763265	192.168.0.41	96.49.215.150	OpenVPN	128	MessageType: P_MESSAGE_V1
54	0.199211616	192.168.0.41	96.49.215.150	OpenVPN	128	MessageType: P_MESSAGE_V1
57	0.644973374	192.168.0.41	96.49.215.150	OpenVPN	144	MessageType: P_MESSAGE_V1

```
Desktop : bash — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View Copy Paste Find
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@fedora Desktop# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.41 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::2981:3f7d%enp0s3 netmask 64 broadcast fe80::1
ether 08:00:27:4b:c3:e0 txqueuelen 1000 (Ethernet)
RX packets 52312 bytes 78580306 (74.9 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 10799 bytes 750097 (732.5 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
```

*tun0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	10.8.0.2	140.211.169.196	TCP	60	60582 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=42318666
2	0.135460476	fe80::a7fc:176b:f19... ff02::2		ICMPv6	48	Router Solicitation
3	0.941753547	10.8.0.2	140.211.169.196	TCP	60	46770 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=42318760
4	1.024766458	10.8.0.2	8.43.85.73	TCP	60	54202 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=19904208
5	1.024777884	10.8.0.2	8.43.85.73	TCP	60	54100 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=19904208
6	1.087988025	10.8.0.2	34.221.3.152	TCP	60	51014 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539680
7	1.984086254	10.8.0.2	140.211.169.196	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 46770 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=42318666
8	2.049458453	10.8.0.2	152.19.134.142	TCP	60	40280 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=78678267
9	4.032710964	10.8.0.2	140.211.169.196	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 46770 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=78678267
10	4.032728589	10.8.0.2	140.211.169.196	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 60582 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=78678267
11	6.464833175	fe80::1365:25ef:cb4... ff02::2		ICMPv6	48	Router Solicitation
12	8.065075206	10.8.0.2	140.211.169.196	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 46770 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539680
13	8.942466800	10.8.0.2	34.221.3.152	TCP	60	38784 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539759
14	9.537351320	10.8.0.2	34.221.3.152	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 51014 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539759
15	9.537376931	10.8.0.2	8.43.85.73	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 54190 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539759
16	9.537380417	10.8.0.2	8.43.85.73	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 54202 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539759
17	9.985339081	10.8.0.2	34.221.3.152	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 38784 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539759
18	10.561180184	10.8.0.2	152.19.134.142	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 40280 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539759
19	12.9326155589	10.8.0.2	34.221.3.152	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 38784 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539759
20	12.096903864	10.8.0.2	140.211.169.196	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 60582 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539759
21	15.485237450	fe80::a7fc:176b:f19... ff02::2		ICMPv6	48	Router Solicitation
22	16.065559682	10.8.0.2	34.221.3.152	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 38784 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539759
23	16.193415520	10.8.0.2	140.211.169.196	TCP	60	[TCP Retransmission] [TCP Port numbers reused] 46770 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=22539759
24	22.337339690	fe80::1365:25ef:cb4... ff02::2		ICMPv6	48	Router Solicitation

The package capture shows that the client (local ip: 192.168.0.41) is talking to the server (public ip: 96.49.215.150) while the connection is on. The other package capture captures the tunnel directly.

Test Case #4: The client's address and geolocation are hidden.

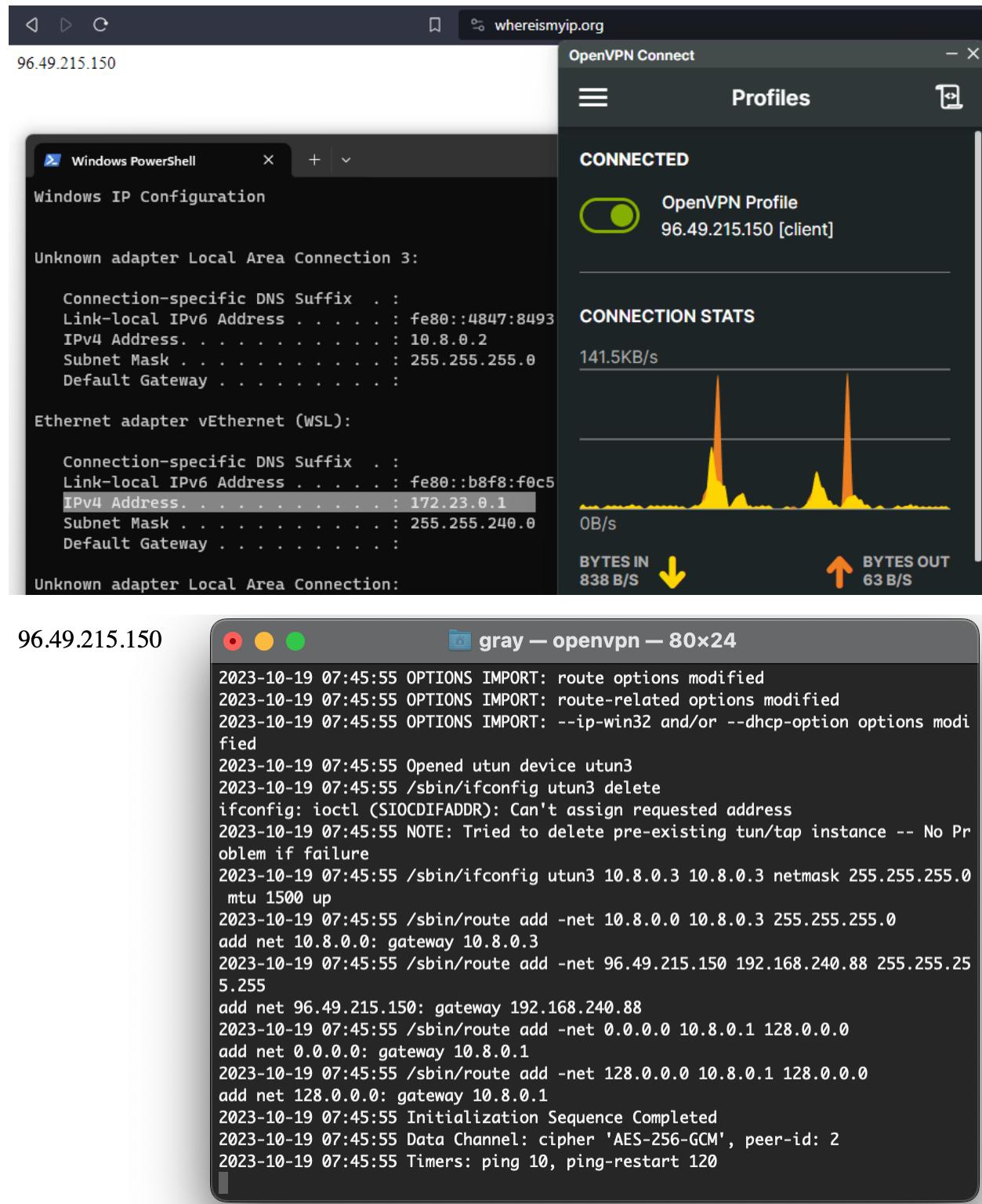
The image contains two screenshots of the OpenVPN Connect application interface, displayed side-by-side in separate browser windows.

Top Screenshot: The title bar shows the IP address "24.114.37.32". The main window title is "OpenVPN Connect" and the sub-section title is "Profiles". It displays the status "DISCONNECTED". An "OpenVPN Profile" section lists the IP address "96.49.215.150 [client]" next to a greyed-out toggle switch icon. A small edit icon is located to the right of the profile name.

Bottom Screenshot: The title bar shows the IP address "96.49.215.150". The main window title is "OpenVPN Connect" and the sub-section title is "Profiles". It displays the status "CONNECTED". The same "OpenVPN Profile" section is shown, but the toggle switch icon is now green and active. Below this, a "CONNECTION STATS" section is partially visible.

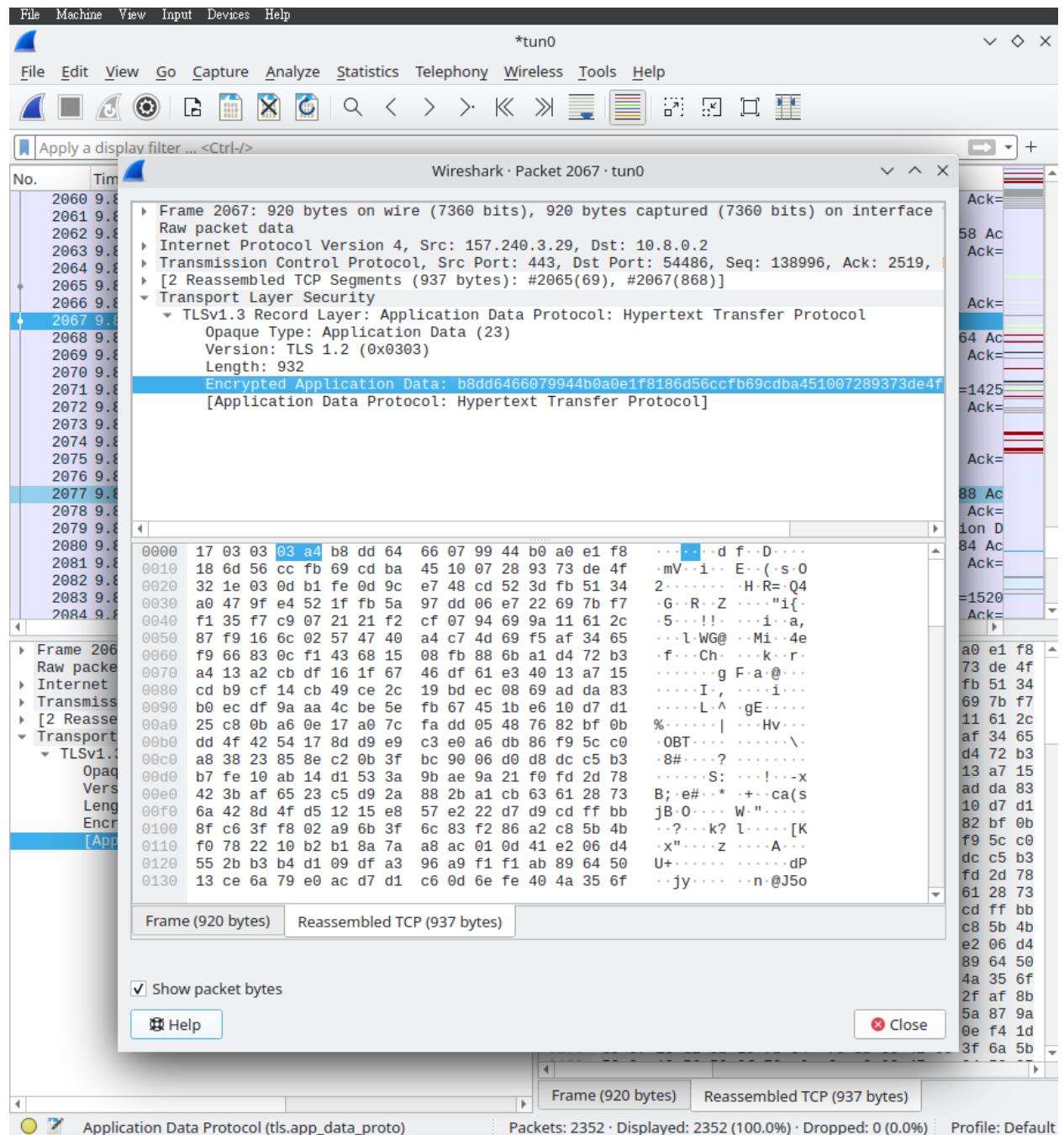
The client's IP address on the internet become the server's address (client is on cellular data).

Test Case #5: Two or more clients can secure their addresses and geolocations.



The public IP addresses on the internet for both clients are all the server's address (both clients are on cellular data: 24.114.38.90)

Test Case #6: Data is secured with encryption.



The capture shows that the traffic in the tunnel contains encrypted tokens.

2.6.2. Post Quantum Algorithm Implementation to VPN

Test Case Number	Description	Status
#1	<p>The server's VPN tunneling functions normally.</p> <p>Pass: The tunnel is opened when the server turns on the VPN, and the tunnel is closed when the server turns off.</p> <p>Fail: The tunnel is either closed while server is on or opened while the server is off.</p>	Pass
#2	<p>The client's VPN tunneling functions normally.</p> <p>Pass: The tunnel is opened when the client connects to the server, and the tunnel is closed when the client does not connect to the server.</p> <p>Fail: The tunnel is either closed while client is connected to the server or opened while the client is not connected to the server.</p>	Pass
#3	<p>Traffic is forwarded through the tunnel to the server.</p> <p>Pass: The server receives the network request.</p> <p>Fail: The server does not receive the network request.</p>	Pass
#4	<p>The client's address and geolocation are hidden.</p> <p>Pass: The IP address on the internet is the server's address.</p> <p>Fail: The IP address on the internet is the client's address.</p>	Pass
#5	<p>Two or more clients can secure their addresses and geolocations.</p> <p>Pass: The IP addresses on the internet for all clients are all the server's address.</p> <p>Fail: One or none of the IP addresses are the server's address.</p>	Pass
#6	<p>Data is quantum safe.</p> <p>Pass: The keys are exchanged with a quantum resistance algorithm.</p> <p>Fail: The keys are exchanged without a quantum resistance algorithm.</p>	Pass

Test Case #1: The server's VPN tunneling functions normally.

The terminal window shows the output of the 'ifconfig' command:

```
[root@fedora Desktop]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.217  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::85e7:c0b:8553:e5cf  prefixlen 64  scopeid 0x20<link>
          ether 08:00:27:04:36:31  txqueuelen 1000  (Ethernet)
            RX packets 302988  bytes 430363682 (410.4 MiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 79191  bytes 6277309 (5.9 MiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
          loop  txqueuelen 1000  (Local Loopback)
            RX packets 782  bytes 208179 (203.2 KiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 782  bytes 208179 (203.2 KiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

[root@fedora Desktop]#
```

The OpenVPN Server configuration window shows the configuration file path set to /root/Desktop/server_test.conf, and the status is Inactive.

The terminal window shows the output of the 'ifconfig' command, including a new tun interface:

```
[root@fedora Desktop]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.217  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::85e7:c0b:8553:e5cf  prefixlen 64  scopeid 0x20<link>
          ether 08:00:27:04:36:31  txqueuelen 1000  (Ethernet)
            RX packets 303182  bytes 430398921 (410.4 MiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 79212  bytes 6279292 (5.9 MiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
          loop  txqueuelen 1000  (Local Loopback)
            RX packets 782  bytes 208179 (203.2 KiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 782  bytes 208179 (203.2 KiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

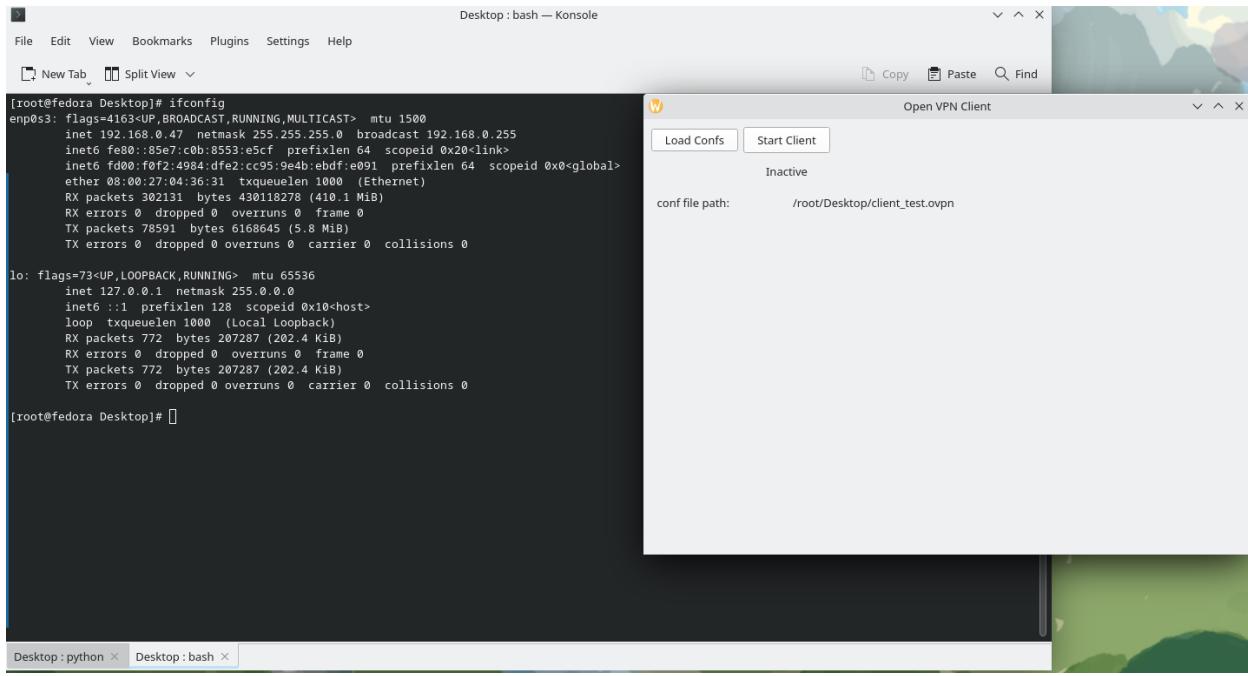
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.0.1  netmask 255.255.255.0  destination 10.8.0.1
        inet6 fe80::9e8e:aec9:76a9:9055  prefixlen 64  scopeid 0x20<link>
          unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
            RX packets 0  bytes 0 (0.0 B)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 6  bytes 356 (356.0 B)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

[root@fedora Desktop]#
```

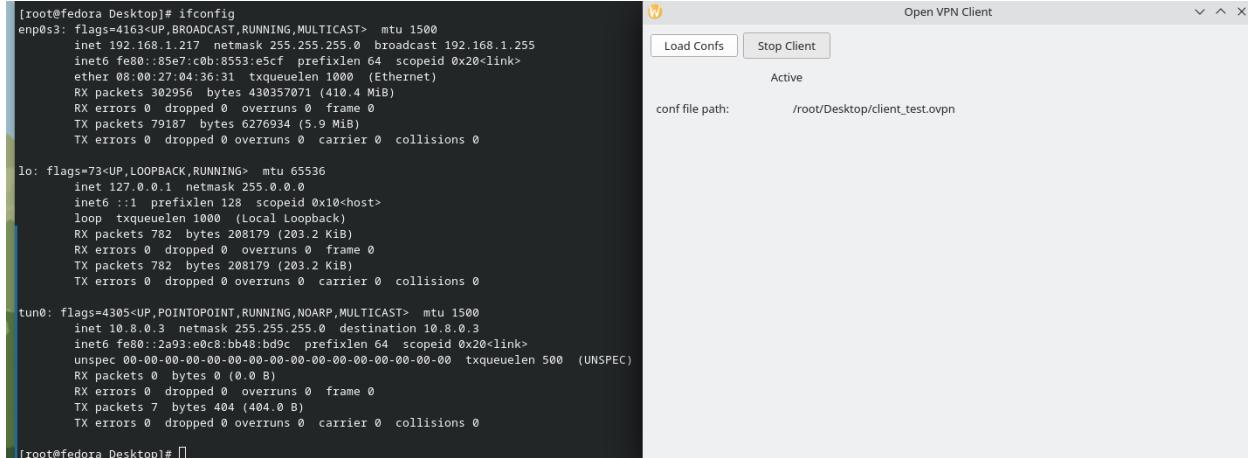
The OpenVPN Server configuration window shows the configuration file path set to /root/Desktop/server_test.conf, and the status is Active.

The tunnel is created while the server is on, and the tunnel is closed after the server is turned off.

Test Case #2: The client's VPN tunneling functions normally.



The screenshot shows a Fedora desktop environment. In the top-left corner, there is a terminal window titled "Desktop : bash — Konsole" with the command "ifconfig" running. The output shows interfaces enp0s3 and lo. In the bottom-left corner, another terminal window titled "Desktop : python" is open. To the right of these windows is an "Open VPN Client" application window. The application has tabs for "Load Conf" and "Start Client". It shows a configuration file path of "/root/Desktop/client_test.ovpn" and indicates the connection status as "Inactive".



The screenshot shows the same Fedora desktop environment. The terminal windows remain the same. The "Open VPN Client" application window now shows the connection status as "Active".

```
[root@fedora Desktop]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.47  netmask 255.255.255.0  broadcast 192.168.0.255
          inet6 fe80::85e7:c0b:8553:ec5f  prefixlen 64  scopeid 0x20<link>
            inet6 fd00:fd02:4984:dfe2::c95:9e4b:ebdf:e091  prefixlen 64  scopeid 0x0<global>
              ether 08:00:27:04:36:31  txqueuelen 1000  (Ethernet)
                RX packets 302131  bytes 43018278 (410.1 MiB)
                RX errors 0  dropped 0  overruns 0  frame 0
                TX packets 78591  bytes 6168645 (5.8 MiB)
                TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
          inet6 ::1  prefixlen 128  scopeid 0x10<host>
            loop  txqueuelen 1000  (Local Loopback)
              RX packets 772  bytes 207287 (202.4 KiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 772  bytes 207287 (202.4 KiB)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

[root@fedora Desktop]# [root@fedora Desktop]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.217  netmask 255.255.255.0  broadcast 192.168.1.255
          inet6 fe80::85e7:c0b:8553:ec5f  prefixlen 64  scopeid 0x20<link>
            ether 08:00:27:04:36:31  txqueuelen 1000  (Ethernet)
              RX packets 302956  bytes 430357071 (410.4 MiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 79187  bytes 6276934 (5.9 MiB)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
          inet6 ::1  prefixlen 128  scopeid 0x10<host>
            loop  txqueuelen 1000  (Local Loopback)
              RX packets 782  bytes 208179 (203.2 KiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 782  bytes 208179 (203.2 KiB)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.0.3  netmask 255.255.255.0  destination 10.8.0.3
          inet6 fe80::2a93:e0c8:bd48:bd9c  prefixlen 64  scopeid 0x20<link>
            unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
              RX packets 0  bytes 0 (0.0 B)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 7  bytes 404 (404.0 B)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

[root@fedora Desktop]# [root@fedora Desktop]#
```

The tunnel is opened when the client connects to the server, and the tunnel is closed when the client does not connect to the server.

Test Case #3: Traffic is forwarded through the tunnel to the server.

Two screenshots of Wireshark showing network traffic captures. The top screenshot shows traffic on interface *tun0 with IP address 10.8.0.3. The bottom screenshot shows traffic on interface 10.8.0.3 and http with IP address 10.8.0.3.

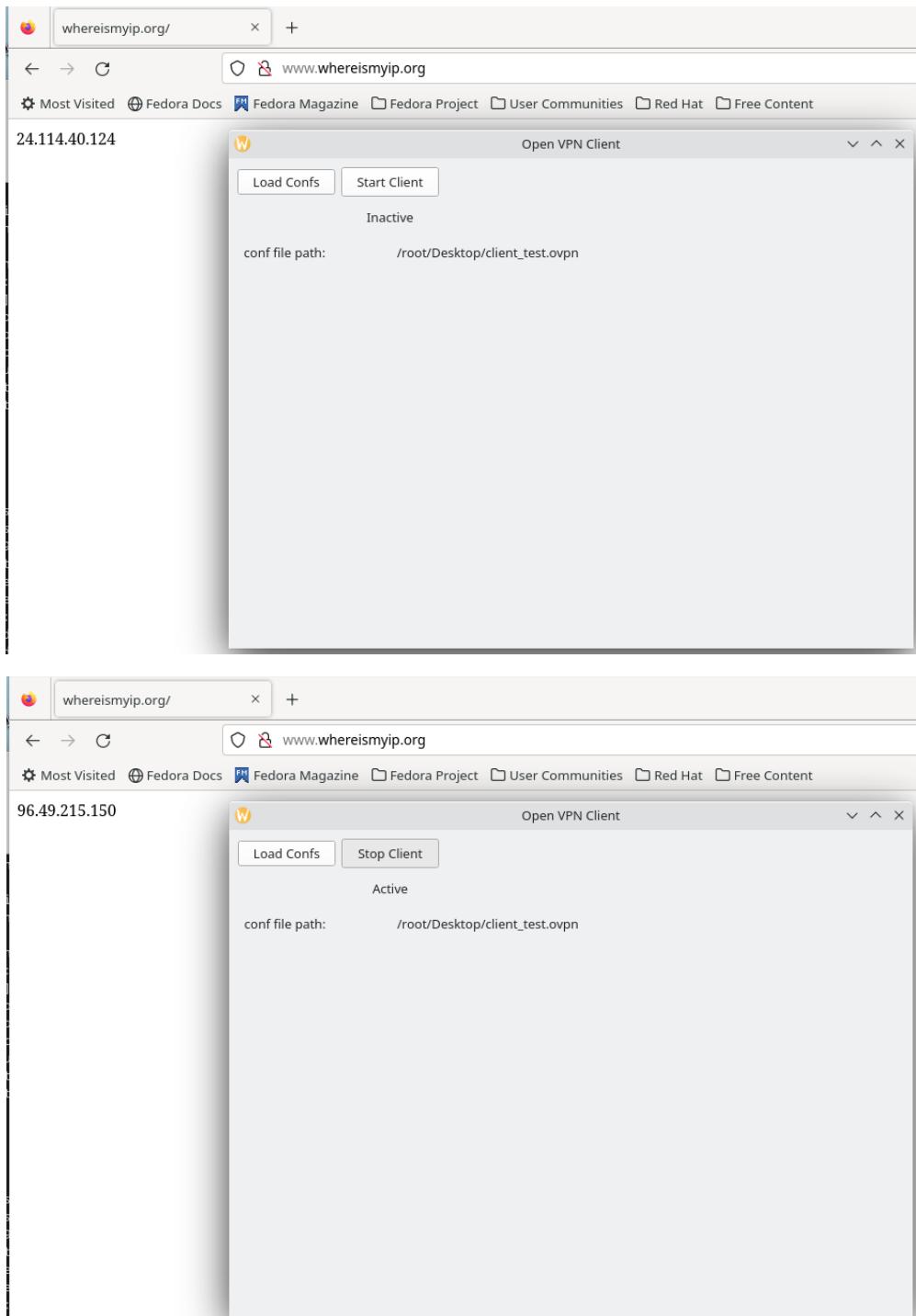
Top Screenshot (Interface *tun0):

No.	Time	Source	Destination	Protocol	Length	Info
8	182.661069582	10.8.0.3	34.160.144.191	TCP	52	43094 -> 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3901572431 TSecr=2302387249
9	182.661548928	10.8.0.3	34.160.144.191	TLSv1.2	264	Client Hello
10	182.677956563	34.160.144.191	10.8.0.3	TCP	52	43094 -> 43994 [ACK] Seq=1 Ack=213 Win=66816 Len=0 TSval=2302387279 TSecr=3901572432
11	182.678841448	10.8.0.3	34.172.170.45	TCP	60	41026 -> 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1399 SACK_PERM TSval=1945185483 TSecr=0 WS=128
12	182.679994236	34.160.144.191	10.8.0.3	TLSv1.2	1439	Server Hello
13	182.680092480	34.160.144.191	10.8.0.3	TCP	1439	443 -> 43994 [PSH, ACK] Seq=1388 Ack=213 Win=66816 Len=1387 TSval=2302387281 TSecr=390157243
14	182.689425455	34.160.144.191	10.8.0.3	TLSv1.2	1439	Certificate
15	182.689425457	34.160.144.191	10.8.0.3	TLSv1.2	294	Server Key Exchange, Server Hello Done
16	182.685897659	10.8.0.3	34.160.144.191	TCP	52	43094 -> 443 [FIN, ACK] Seq=213 Ack=1 Win=64256 Len=0 TSval=3901572450 TSecr=2302387249
17	182.687010798	10.8.0.3	34.160.144.191	TCP	40	43094 -> 443 [RST] Seq=213 Win=0 Len=0
18	182.687038750	10.8.0.3	34.160.144.191	TCP	40	43094 -> 443 [RST] Seq=213 Win=0 Len=0
19	182.687054279	10.8.0.3	34.160.144.191	TCP	40	43094 -> 443 [RST] Seq=213 Win=0 Len=0
20	182.687087063	10.8.0.3	34.160.144.191	TCP	40	43094 -> 443 [RST] Seq=213 Win=0 Len=0
21	182.697483353	10.8.0.3	34.172.170.45	TCP	60	41026 [SYN] ACK Seq=214 Win=65535 Len=0 MSS=1440 SACK_PERM TSval=4049767183 TSecr=3901572450
22	182.699517265	10.8.0.3	34.172.170.45	TCP	40	41026 -> 443 [RST] Seq=1 Win=0 Len=0
23	182.699517269	34.160.144.191	10.8.0.3	TCP	52	43094 -> 443 [SYN] ACK Seq=1403 Win=214 Win=66816 Len=0 TSval=2302387303 TSecr=3901572450
24	182.705998955	10.8.0.3	34.160.144.191	TCP	40	43094 -> 443 [RST] Seq=214 Win=0 Len=0
25	182.845930378	10.8.0.3	34.197.221.82	TCP	60	55540 -> 80 [SYN] Seq=1 Win=64240 Len=0 MSS=1399 SACK_PERM TSval=579975109 TSecr=0 WS=128
26	182.862064075	34.160.144.191	10.8.0.3	TCP	60	80 [SYN] ACK Seq=1 Win=65535 Len=0 MSS=1412 SACK_PERM TSval=3354972712 TSecr=0 WS=128
27	182.887207044	10.8.0.3	34.197.221.82	TCP	52	55540 -> 80 [ACK] Seq=1 Win=64256 Len=0 TSval=579975138 TSecr=3354972712
28	182.887249159	10.8.0.3	34.197.221.82	HTTP	357	GET /canonical.html HTTP/1.1
29	182.913929399	10.8.0.3	34.117.237.239	TCP	60	48142 -> 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1399 SACK_PERM TSval=2289874520 TSecr=0 WS=128
30	183.018235640	34.160.144.191	10.8.0.3	TCP	52	80 [SYN] ACK Seq=1 Win=66816 Len=0 TSval=3354972750 TSecr=579975138
31	183.018394479	34.160.144.191	10.8.0.3	HTTP	350	HTTP/1.1 200 [text/html]
32	183.021496597	10.8.0.3	34.197.221.82	TCP	52	55540 -> 80 [ACK] Seq=396 Ack=299 Win=64128 Len=0 TSval=579975286 TSecr=3354972751

Bottom Screenshot (Interface 10.8.0.3 and http):

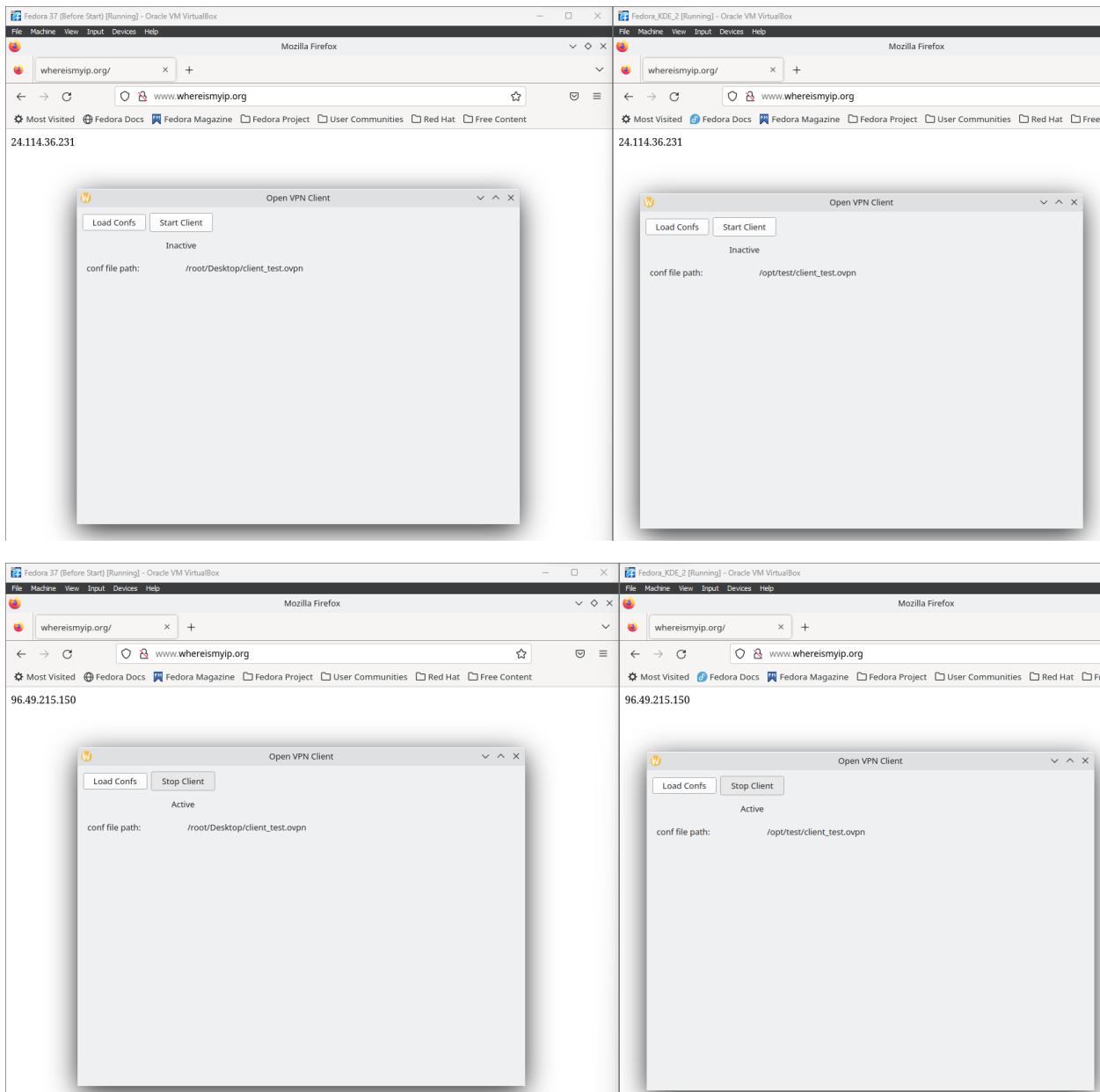
No.	Time	Source	Destination	Protocol	Length	Info																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
28	182.882748169	10.8.0.3	34.197.221.82	HTTP	357	GET /canonical.html HTTP/1.1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
29	183.018394479	10.8.0.3	34.197.221.82	HTTP	650	HTTP/1.1 200 [text/html]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
30	183.018394479	10.8.0.3	34.197.221.82	HTTP	350	GET /success.txt?ipv4 HTTP/1.1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
31	183.018394479	10.8.0.3	34.197.221.82	HTTP	268	HTTP/1.1 200 [text/plain]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
32	183.018394479	10.8.0.3	34.197.221.82	HTTP	469	HTTP/1.1 200 [text/plain]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
33	183.018394479	10.8.0.3	34.197.221.82	HTTP	789	HTTP/1.1 200 [text/plain]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
34	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
35	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
36	183.018394479	10.8.0.3	34.197.221.82	HTTP	483	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
37	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
38	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
39	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
40	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
41	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
42	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
43	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
44	183.018394479	10.8.0.3	34.197.221.82	HTTP	224	22.24	45	183.018394479	10.8.0.3	34.197.221.82	HTTP	941	Response	46	183.018394479	10.8.0.3	34.197.221.82	HTTP	483	Request	47	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	48	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	49	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	50	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	51	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	52	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	53	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	54	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	55	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	56	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	57	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	58	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	59	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	60	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	61	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	62	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	63	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	64	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	65	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	66	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	67	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	68	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	69	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	70	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	71	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	72	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	73	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	74	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	75	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	76	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	77	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	78	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	79	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	80	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	81	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	82	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	83	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	84	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	85	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	86	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	87	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	88	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	89	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	90	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	91	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	92	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	93	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	94	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	95	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	96	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	97	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	98	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	99	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	100	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	101	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	102	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	103	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	104	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	105	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	106	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	107	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	108	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	109	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	110	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	111	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	112	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	113	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	114	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	115	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	116	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	117	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	118	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	119	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	120	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	121	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	122	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	123	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	124	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	125	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	126	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	127	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response	128	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	129	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response	130	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request	131	183.018
45	183.018394479	10.8.0.3	34.197.221.82	HTTP	941	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
46	183.018394479	10.8.0.3	34.197.221.82	HTTP	483	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
47	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
48	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
49	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
50	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
51	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
52	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
53	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
54	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
55	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
56	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
57	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
58	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
59	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
60	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
61	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
62	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
63	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
64	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
65	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
66	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
67	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
68	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
69	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
70	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
71	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
72	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
73	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
74	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
75	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
76	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
77	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
78	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
79	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
80	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
81	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
82	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
83	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
84	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
85	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
86	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
87	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
88	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
89	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
90	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
91	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
92	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
93	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
94	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
95	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
96	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
97	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
98	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
99	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
100	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
101	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
102	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
103	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
104	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
105	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
106	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
107	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
108	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
109	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
110	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
111	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
112	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
113	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
114	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
115	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
116	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
117	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
118	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
119	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
120	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
121	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
122	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
123	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
124	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
125	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
126	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
127	183.018394479	10.8.0.3	34.197.221.82	HTTP	754	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
128	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
129	183.018394479	10.8.0.3	34.197.221.82	HTTP	753	Response																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
130	183.018394479	10.8.0.3	34.197.221.82	HTTP	482	Request																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
131	183.018																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	

Test Case #4: The client's address and geolocation are hidden.



The client's public IP address on the internet becomes the server's address (client is on cellular data).

Test Case #5: Two or more clients can secure their addresses and geolocations.



The public IP addresses on the internet for both clients are all the server's address (both clients are on cellular data)

Test Case #6: Data is quantum safe.

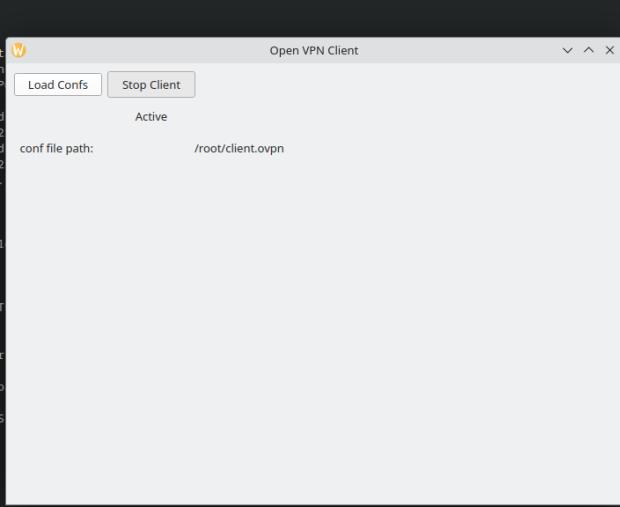
udp.port==1194						
No.	Time	Source	Destination	Protocol	Length	Info
47	2.038716240	96.49.215.150	192.168.0.41	OpenVPN	62	MessageType: P_CONTROL_HARD_RESET_CLIENT_V2
48	2.039056988	192.168.0.41	96.49.215.150	OpenVPN	70	MessageType: P_CONTROL_HARD_RESET_SERVER_V2
49	2.045075972	96.49.215.150	192.168.0.41	TLSv1.3	1266	
50	2.045076509	96.49.215.150	192.168.0.41	TLSv1.3	329	Continuation Data
51	2.045949984	192.168.0.41	96.49.215.150	OpenVPN	66	MessageType: P_ACK_V1
52	2.047316014	192.168.0.41	96.49.215.150	TLSv1.3	1266	Server Hello, Change Cipher Spec
53	2.047360859	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data
54	2.047449805	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data
55	2.047454752	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data
56	2.047455289	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data
57	2.047455931	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data
58	2.051090585	96.49.215.150	192.168.0.41	OpenVPN	70	MessageType: P_ACK_V1
59	2.051280767	192.168.0.41	96.49.215.150	TLSv1.3	1266	Continuation Data
60	2.052265603	96.49.215.150	192.168.0.41	OpenVPN	74	MessageType: P_ACK_V1
61	2.052265772	96.49.215.150	192.168.0.41	OpenVPN	78	MessageType: P_ACK_V1
62	2.052265795	96.49.215.150	192.168.0.41	OpenVPN	78	MessageType: P_ACK_V1
63	2.052265817	96.49.215.150	192.168.0.41	OpenVPN	78	MessageType: P_ACK_V1
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)						
Compression Method: null (0)						
Extensions Length: 1102						
▼ Extension: supported_versions (len=2)						
Type: supported_versions (43)						
Length: 2						
Supported Version: TLS 1.3 (0x0304)						
▼ Extension: key_share (len=1092)						
Type: key_share (51)						
Length: 1092						
▼ Key Share extension						
▼ Key Share Entry: Group: kyber768, Key Exchange length: 1088						
Group: kyber768 (572)						
Key Exchange Length: 1088						
Key Exchange: 91f1df2665bb40f1864260bdf8288012a295019b821fdafb870f631636e11a565b5ad96b..						
▼ TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec						
Content Type: Change Cipher Spec (20)						
Version: TLS 1.2 (0x0303)						

The keys are exchanged with the kyber768 algorithm when the connection starts.

2.6.3. GUI Integration

Test Case Number	Description	Status
#1	<p>Client can connect to the server without crashing.</p> <p>Pass: GUI successfully responds to the client and connects to the server. Fail: GUI does not respond to the client or crashes.</p>	Pass
#2	<p>Server can start the service without crashes.</p> <p>Pass: GUI successfully responds to the server and starts the application. Fail: GUI does not respond to the server or crashes if port is currently used.</p>	Pass
#3	<p>Client selects other files that are not with the extension .ovpn.</p> <p>Pass: GUI does not allow the client to select files that do not end in .ovpn extension. Fail: GUI allows the client to select files that do not end in .ovpn extension.</p>	Pass
#4	<p>Server selects other files that are not with the extension .conf.</p> <p>Pass: GUI does not allow the server to select files that do not end in .conf extension. Fail: GUI allows the server to select files that do not end in .conf extension.</p>	Pass
#5	<p>Traffic is collected in a log file.</p> <p>Pass: All traffic is written in a log file. Fail: Traffic is not logged in a file.</p>	Pass

Test Case #1: Client can connect to the server without crashing.

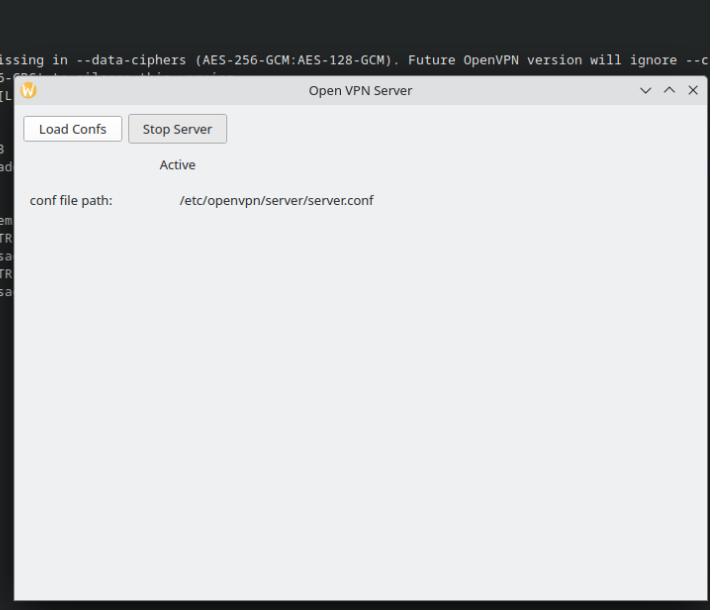


The screenshot shows the 'Open VPN Client' window. At the top, there are two buttons: 'Load Conf' and 'Stop Client'. Below them, the status is 'Active'. A configuration file path is listed as '/root/client.ovpn'. The main area of the window displays a terminal log of the client's connection process, which ends with a success message indicating the connection is established.

```
sh-5.2# python3 openvpn_client.py
VMware: No 3D enabled (0, Success).
libEGL warning: egl: failed to create dri2 screen
2023-10-19 01:36:01 DEPRECATED OPTION: --cipher set to 'AES-256-CBC' but missing in --data-ciphers or change --cipher 'AES-256-CBC' to --data-ciphers-fallback 'AES-256-CBC' to silence this warning
2023-10-19 01:36:01 OpenVPN 2.5.9 x86_64-redhat-linux-gnu [SSL (OpenSSL)] [LZO] [L4] [EPOLL] [AES-256]
2023-10-19 01:36:01 library versions: OpenSSL 3.0.8 7 Feb 2023, LZO 2.10
2023-10-19 01:36:01 Outgoing Control Channel Encryption: Cipher 'AES-256-CTR' initialized
2023-10-19 01:36:01 Incoming Control Channel Encryption: Using 256 bit message hash 'SHA256'
2023-10-19 01:36:01 Incoming Control Channel Encryption: Cipher 'AES-256-CTR' initialized
2023-10-19 01:36:01 Incoming Control Channel Encryption: Using 256 bit message hash 'SHA256'
2023-10-19 01:36:01 TCP/UDP: Preserving recently used remote address: [AF_INET]96.49.215.150
2023-10-19 01:36:01 Socket Buffers: R=[212992->212992] S=[212992->212992]
2023-10-19 01:36:01 UDP link local: (not bound)
2023-10-19 01:36:01 UDP link remote: [AF_INET]96.49.215.150:1194
2023-10-19 01:36:01 TLS: Initial packet from [AF_INET]96.49.215.150:1194, sid=673f6f5b 01:36:01 VERIFY OK: depth=1, CN=Easy-RSA CA
2023-10-19 01:36:01 VERIFY KU OK
2023-10-19 01:36:01 Validating certificate extended key usage
2023-10-19 01:36:01 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
2023-10-19 01:36:01 VERIFY EKU OK
2023-10-19 01:36:01 VERIFY OK: depth=0, CN=server
2023-10-19 01:36:01 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, peer [AF_INET]96.49.215.150:1194
2023-10-19 01:36:01 [server] Peer Connection Initiated with [AF_INET]96.49.215.150:1194
2023-10-19 01:36:01 PUSH: Received control message: 'PUSH_REPLY,redirect-gateway bypass 2.255.255.0,peer-id 0,cipher AES-256-GCM'
2023-10-19 01:36:01 Unrecognized option or missing or extra parameter(s) in [PUSH-OPTIONS]
2023-10-19 01:36:01 OPTIONS IMPORT: timers and/or timeouts modified
2023-10-19 01:36:01 OPTIONS IMPORT: --ifconfig/up options modified
2023-10-19 01:36:01 OPTIONS IMPORT: route options modified
2023-10-19 01:36:01 OPTIONS IMPORT: route-related options modified
2023-10-19 01:36:01 OPTIONS IMPORT: --ip-win32 and/or --dhcp-option options modified
```

GUI successfully responds to the client and connects to the server.

Test Case #2: Server can start the service without crashes.

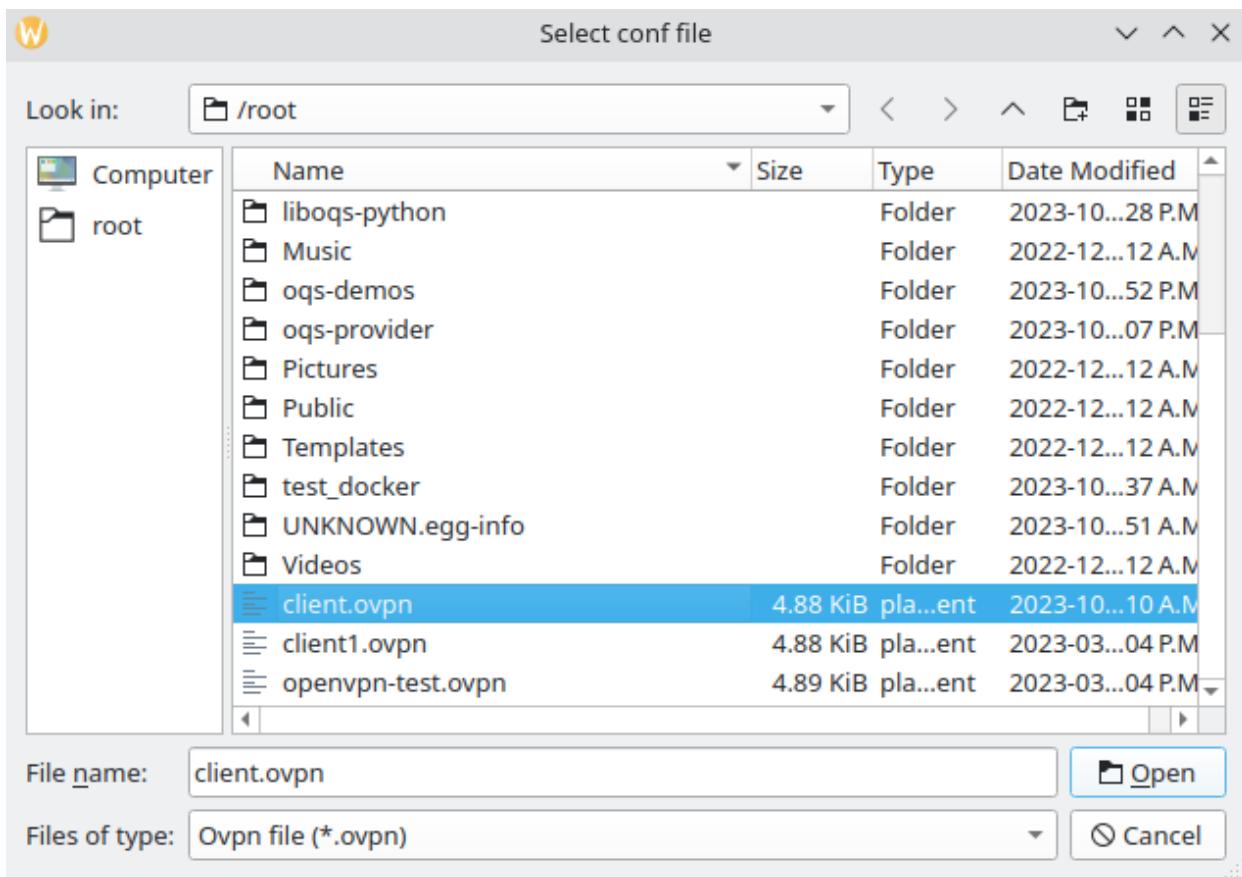


The screenshot shows the 'Open VPN Server' window. At the top, there are two buttons: 'Load Conf' and 'Stop Server'. Below them, the status is 'Active'. A configuration file path is listed as '/etc/openvpn/server/server.conf'. The main area of the window displays a terminal log of the server's startup process, which ends with a success message indicating the initialization sequence completed.

```
sh-5.2# python3 openvpn_server.py
VMware: No 3D enabled (0, Success).
libEGL warning: egl: failed to create dri2 screen
2023-10-19 01:14:25 DEPRECATED OPTION: --cipher set to 'AES-256-CBC' but missing in --data-ciphers (AES-256-GCM:AES-128-GCM). Future OpenVPN version will ignore --cipher or change --cipher 'AES-256-CBC' to --data-ciphers-fallback 'AES-256-CBC' to silence this warning
2023-10-19 01:14:25 OpenVPN 2.5.9 x86_64-redhat-linux-gnu [SSL (OpenSSL)] [LZO] [L4] [EPOLL] [AES-256]
2023-10-19 01:14:25 library versions: OpenSSL 3.0.8 7 Feb 2023, LZO 2.10
2023-10-19 01:14:25 net_route_v4_best_gw query: dst 0.0.0.0
2023-10-19 01:14:25 net_route_v4_best_gw result: via 192.168.0.1 dev enp0s3
2023-10-19 01:14:25 NOTE: your local LAN uses the extremely common subnet address 192.168.0.0/16, which may cause problems with other
internet cafes that use the same subnet.
2023-10-19 01:14:25 Diffie-Hellman initialized with 2048 bit key
2023-10-19 01:14:25 CRL: loaded 1 CRLs from file /etc/openvpn/server/crl.pem
2023-10-19 01:14:25 Outgoing Control Channel Encryption: Cipher 'AES-256-CTR' initialized
2023-10-19 01:14:25 Outgoing Control Channel Encryption: Using 256 bit message hash 'SHA256'
2023-10-19 01:14:25 Incoming Control Channel Encryption: Cipher 'AES-256-CTR' initialized
2023-10-19 01:14:25 Incoming Control Channel Encryption: Using 256 bit message hash 'SHA256'
2023-10-19 01:14:25 TUN/TAP device tun0 opened
2023-10-19 01:14:25 net_iface_mtu_set: mtu 1500 for tun0
2023-10-19 01:14:25 net_iface_up: set tun0 up
2023-10-19 01:14:25 net_addr_v4_add: 10.8.0.1/24 dev tun0
2023-10-19 01:14:25 Could not determine IPv4/IPv6 protocol. Using AF_INET
2023-10-19 01:14:25 Socket Buffers: R=[212992->212992] S=[212992->212992]
2023-10-19 01:14:25 UDPv4 link local (bound): [AF_INET]192.168.0.37:1194
2023-10-19 01:14:25 UDPv4 link remote: [AF_UNSPEC]
2023-10-19 01:14:25 GID set to nobody
2023-10-19 01:14:25 UID set to nobody
2023-10-19 01:14:25 MULTI: multi_init called, r=256 v=256
2023-10-19 01:14:25 IFCONFIG POOL IPv4: base=10.8.0.2 size=253
2023-10-19 01:14:25 ifconfig_pool_l_read(): in='client,10.8.0.2,'
2023-10-19 01:14:25 succeeded -> ifconfig_pool_set(hand=0)
2023-10-19 01:14:25 IFCONFIG POOL LIST
2023-10-19 01:14:25 client,10.8.0.2,
2023-10-19 01:14:25 Initialization Sequence Completed
```

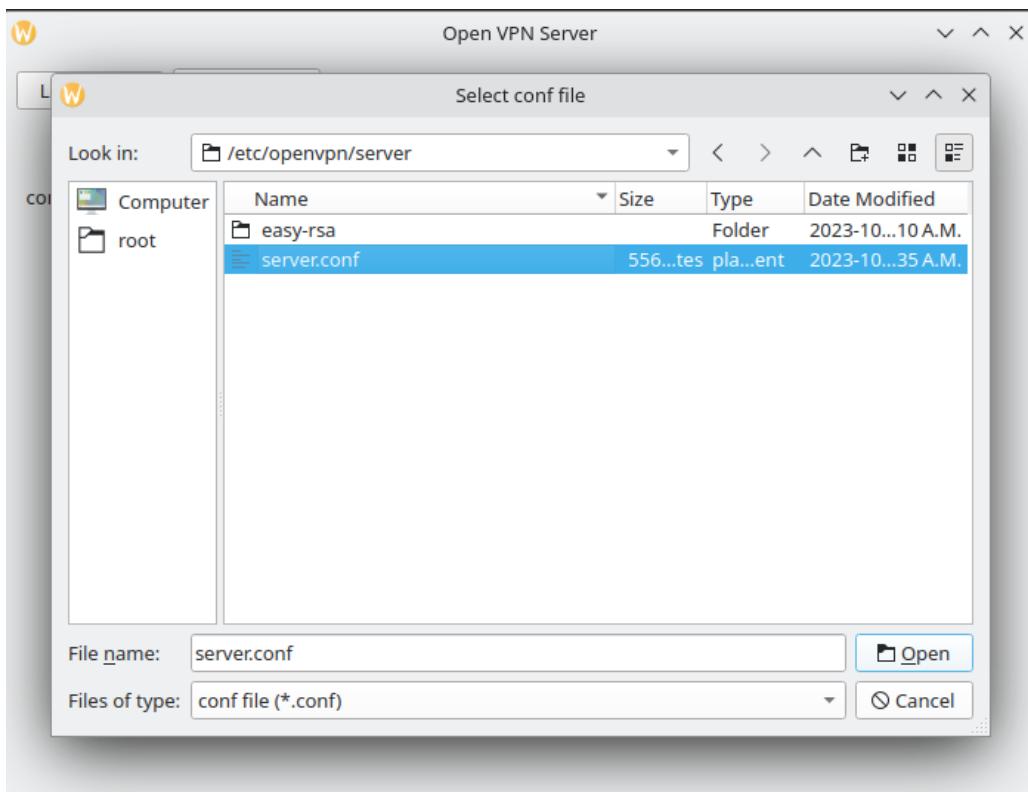
GUI successfully responds to the server and starts the application.

Test Case #3: Client selects other files that are not with the extension .ovpn.



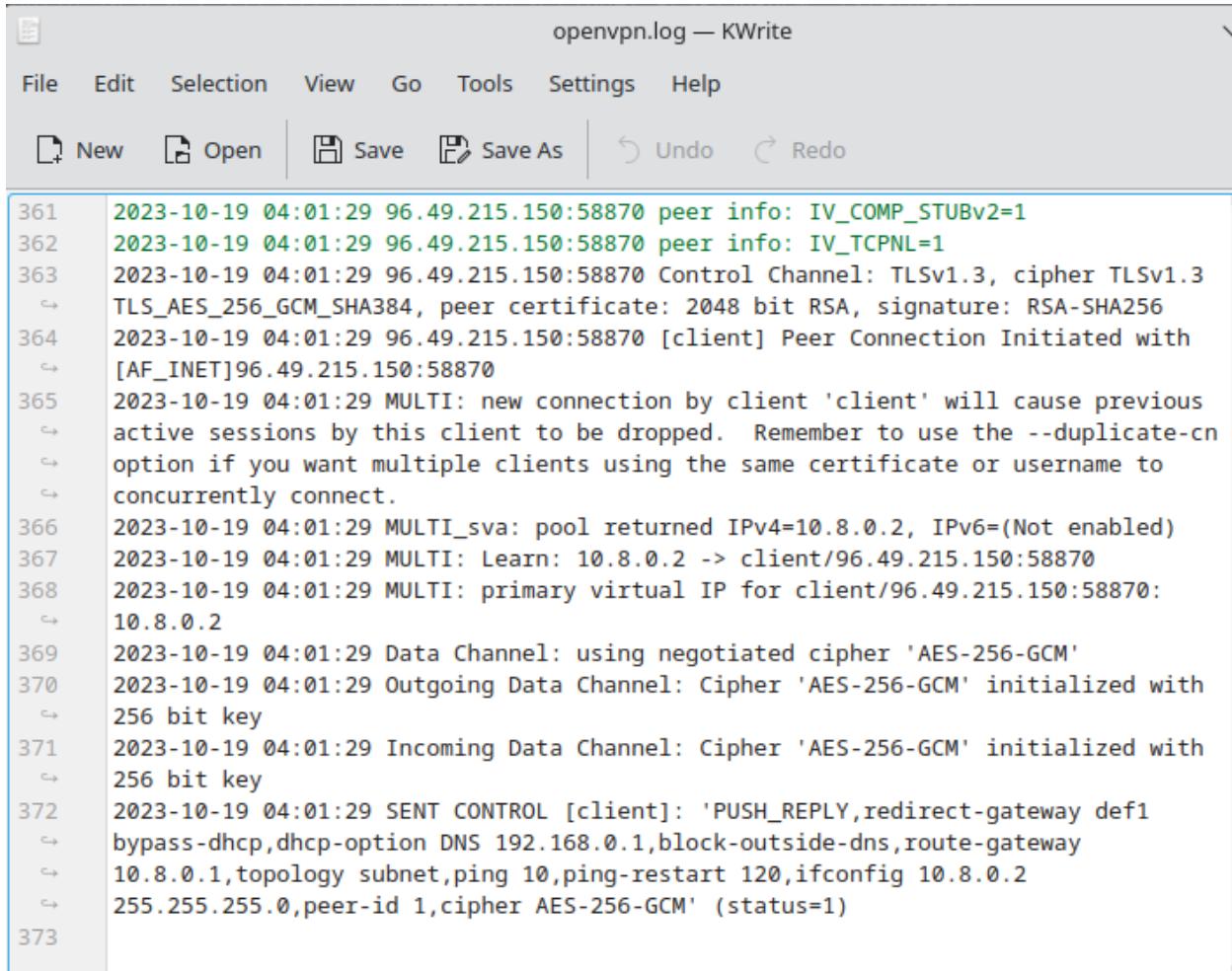
GUI does not allow the client to select files that do not end in .ovpn extension.

Test Case #4: Server selects other files that are not with the extension .conf.



GUI does not allow the server to select files that do not end in .conf extension.

Test Case #5: Traffic is collected in a log file.



The screenshot shows a KWrite text editor window titled "openvpn.log — KWrite". The menu bar includes File, Edit, Selection, View, Go, Tools, Settings, and Help. Below the menu is a toolbar with New, Open, Save, Save As, Undo, and Redo buttons. The main text area displays log entries from line 361 to 373. The log entries detail the negotiation and establishment of a VPN connection between a client and a server, including TLS handshake details, cipher selection (AES-256-GCM), and configuration parameters like topology, ping, and route settings.

```
361 2023-10-19 04:01:29 96.49.215.150:58870 peer info: IV_COMP_STUBv2=1
362 2023-10-19 04:01:29 96.49.215.150:58870 peer info: IV_TCPNL=1
363 2023-10-19 04:01:29 96.49.215.150:58870 Control Channel: TLSv1.3, cipher TLSv1.3
  ↳ TLS_AES_256_GCM_SHA384, peer certificate: 2048 bit RSA, signature: RSA-SHA256
364 2023-10-19 04:01:29 96.49.215.150:58870 [client] Peer Connection Initiated with
  ↳ [AF_INET]96.49.215.150:58870
365 2023-10-19 04:01:29 MULTI: new connection by client 'client' will cause previous
  ↳ active sessions by this client to be dropped. Remember to use the --duplicate-cn
  ↳ option if you want multiple clients using the same certificate or username to
  ↳ concurrently connect.
366 2023-10-19 04:01:29 MULTI_sva: pool returned IPv4=10.8.0.2, IPv6=(Not enabled)
367 2023-10-19 04:01:29 MULTI: Learn: 10.8.0.2 -> client/96.49.215.150:58870
368 2023-10-19 04:01:29 MULTI: primary virtual IP for client/96.49.215.150:58870:
  ↳ 10.8.0.2
369 2023-10-19 04:01:29 Data Channel: using negotiated cipher 'AES-256-GCM'
370 2023-10-19 04:01:29 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with
  ↳ 256 bit key
371 2023-10-19 04:01:29 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with
  ↳ 256 bit key
372 2023-10-19 04:01:29 SENT CONTROL [client]: 'PUSH_REPLY,redirect-gateway def1
  ↳ bypass-dhcp,dhcp-option DNS 192.168.0.1,block-outside-dns,route-gateway
  ↳ 10.8.0.1,topology subnet,ping 10,ping-restart 120,ifconfig 10.8.0.2
  ↳ 255.255.255.0,peer-id 1,cipher AES-256-GCM' (status=1)
373
```

All traffic is written in a log file when the VPN is configured to capture logs in a separate file.

2.7. Implications of Implementation

Implications of the implementation on my Quantum-Resistant VPN project are complicated. Initially, my limited knowledge of the quantum realm and post-quantum cryptography posed a significant challenge. I dedicated a substantial amount of time to researching and exploring the libraries offered by the Open Quantum Safe Project, aiming to ensure that the VPN remains secure against emerging quantum threats. I thought that the OpenVPN protocol did not inherently support quantum-safe algorithms, but I was wrong after closely examining the smaller part of the protocol, the OpenSSL. After changing the version of OpenSSL to the latest and rebuilding OpenVPN with it, the application is finally able to recognize the certificates generated by post quantum algorithm.

Also, I was pleased to find that the development of the Graphical User Interface (GUI) progressed more swiftly than initially anticipated. Notably, I got an opportunity to implement additional requirements, which is the implementation of another vital component of post-quantum cryptography: the Signature Scheme. This additional implementation enhances the project's versatility and positions it favorably for future developments in post-quantum cryptography and quantum-resistant security solutions.

2.8. Innovation

The Open Quantum Safe (OQS) project that aims to create quantum-resistant cryptography was started in late 2016, which is a relatively new concept in cryptography, so its implementation of it should be considered innovative as it is considered a future leading-edge algorithm. My application not only implements cryptography but also integrates the encryption algorithms with the SSL protocol that is used in OpenVPN. The implementation creates an entirely new VPN application that is well-prepared for future attacks from both classical and quantum computers.

One of the similar products to my application is Mullvad VPN. It is a VPN application that also implements post-quantum cryptography. It uses WireGuard as its communication protocol and classic McEliece as its post-quantum encryption algorithms (Mullvad, 2022). Since WireGuard is written in around 4000 to 5000 lines, it does not provide much freedom when it comes to encryption and security compared to OpenVPN, which is written in at least 70,000 lines. If we are to implement a new algorithm in our application, WireGuard would have more restrictions when it comes to development.

2.9. Complexity

The complexity of this project depends on the newly introduced cryptography and the integration with an existing VPN protocol. Because post-quantum cryptography is relatively new, it will be constantly changing when the developer implements it. Since the liboqs library, which is an open-source library for quantum-safe cryptographic algorithms, only provides algorithms that are accepted by NIST, students will need to implement the entire encryption process manually, including generating and validating keys by using the post quantum algorithms. Diploma students will not be able to solve this problem because they have no knowledge of cryptography and very limited knowledge of VPN protocols. This project requires enough knowledge of networking and application protocols to be able to complete it. Also, for the OpenVPN protocol, students will need to know how SSL (Secure Socket Layer) protocol works, as well as the knowledge of encryption.

2.10. Research in New Technologies

The following is a list of new technologies that I researched or implemented:

- **Modern Cryptology Knowledge:** Before delving into cryptography, I completed a cryptology course at BCIT to establish a fundamental understanding of the subject. This course introduced me to contemporary encryption algorithms such as AES, Triple DES, RSA, and more. It provided the cornerstone for my cryptography knowledge, proving invaluable as I frequently encountered these modern encryption algorithms during my exploration of new technologies.
- **Post Quantum Cryptography:** This is an entirely novel field of study. I invested a substantial amount of time comprehending its nuances, including how it differs from modern cryptography. I also delved into the extensive libraries offered by the Open Quantum Safe project. This was a valuable opportunity to explore this emerging field of cryptology.

- **OpenSSL:** OpenSSL plays a pivotal role in this project since the OpenVPN protocol relies on it for encryption and hashing algorithms. Throughout the project, I successfully generated quantum-resistant certificates using the Dilithium3 algorithm via OpenSSL with the OQS provider and used them for signature verification.
- **GUI Library:** Exploring the PyQt library proved to be a rewarding experience, primarily due to its clear documentation. This library offers comprehensive support for a wide range of elements and includes prebuilt basic layouts. For instance, implementing a file selection window is straightforward compared to other GUI libraries. Moreover, it facilitates the application of filters for file extensions with minimal code. However, PyQt's products tend to share a similar appearance, which presented challenges in aligning them with my design vision, ultimately necessitating some adjustments. Overall, PyQt is an excellent choice for initiating a basic application with a user-friendly interface.
- **VPN Protocol:** During the project, I delved into the intricacies of the OpenVPN protocol. Significant time was dedicated to configuring my application in tandem with this protocol, which boasts a wide array of configuration options. However, a thorough understanding of firewall and iptables rules is imperative when configuring specific options, particularly when routing all client traffic through the server. Although I underestimated the complexity, I successfully created a VPN application that uses quantum resistance algorithms with the OpenVPN Protocol.

2.11. Future Enhancements

The following list outlines several enhancements that can be implemented in the future for this project:

- **Cross-Platform Functionality for the VPN Application:** Expanding the VPN application's compatibility to run efficiently across various operating systems, ensuring accessibility for a wider range of users.
- **Support for Multiple Protocols:** Enhancing the application's versatility by enabling it to handle diverse protocols beyond OpenVPN, accommodating a broader spectrum of user needs.
- **Selective Traffic Forwarding:** Implementing the capability to route network traffic solely from specific applications, offering fine-grained control and network efficiency.
- **Detection of Intrusion Attempts:** Integrating advanced security measures to identify potential attacks, including the detection of port scanning or spoofing attempts.
- **User Preferences and Configuration Saving:** Facilitating a more user-friendly experience by enabling the application to save user preferences and configurations, streamlining the setup and operation process.
- **Performance Optimization:** Focusing on improving performance to reduce packet loss or latency when connecting to multiple clients, ensuring a seamless and responsive user experience.

2.12. Timeline and Milestones

Task	Estimated Hours	Total per Milestone	Actual Hours
Project Initialization			
Setup GitHub Repository	1 hours	5 hours	1 hour
Setup New Project (Pycharm)	1 hours		1 hour
Import Libraries (liboqs, OpenVPN)	3 hours		10 hours
Milestone 1: VPN with OpenVPN Protocol			
Researching basic VPN implementation with OpenVPN Protocol	15 hours	100 hours	20 hours
Planning and Gathering Requirements	10 hours		15 hours
Designing architecture of the VPN application	15 hours		15 hours
Implementation	30 hours		60 hours
Writing and Running Tests	20 hours		20 hours
Documentation	10 hours		10 hours
Milestone 2: Post Quantum Algorithm Implementation			
Researching and learning about encryption algorithms	30 hours	175 hours	30 hours
Exploring liboqs library	15 hours		30 hours
Exploring CRYSTALS-KYBER Algorithm	15 hours		30 hours
Designing architecture	15 hours		15 hours
Implementation	60 hours		120 hours
Writing and Running Tests	30 hours		30 hours
Documentation	10 hours		10 hours

<i>Milestone 3: GUI Integration</i>			
Exploring PyQt library	10 hours	85 hours	10 hours
Designing/Drawing wireframes for both applications (Client and Server)	5 hours		5 hours
Building basic elements for home/main page of the application	10 hours		10 hours
Integrating functions into GUI	30 hours		10 hours
Writing and Running Tests	20 hours		10 hours
Documentation	10 hours		10 hours
<i>Milestone 4: Final Documentation</i>			
Finishing and revising the Final Report	20 hours	30 hours	30 hours
Gathering files for deliverables	10 hours		10 hours
Total:		395 hours	512 hours

Milestone	Completion Date
Milestone 1: VPN with OpenVPN Protocol	March 20, 2023
Milestone 2: Post Quantum Algorithm Implementation	August 24, 2023
Milestone 3: GUI Integration	September 30, 2023
Milestone 4: Final Documentation	October 18, 2023

As highlighted in the development process, Milestone 2 required a considerably greater amount of time than initially anticipated. The underestimation of the complexity of integrating post-quantum cryptography into the project was a significant factor. The liboqs-python library, an essential component of the Open Quantum Safe Project, presented limitations in terms of functionality. This, in turn, posed challenges in implementing the desired quantum-resistant algorithms seamlessly. The cumulative effect

of these challenges resulted in an extension of the project's timeline, surpassing the initial estimated deadlines outlined in the project proposal. While there have been numerous obstacles when exploring the field of quantum cryptography, I remain resolute in my determination to complete the entire project.

3. Conclusion

Unlike the class assignments and projects, this project aims to extend on the development of VPN application, which is a perfect project for a BTech student in the Network Security Applications Development Option. It involves implementing VPN with a well-known and well-used protocol, OpenVPN. Also, it requires the understanding of cryptography to implement the algorithms of encryption for post-quantum purpose. It is a great opportunity to gain knowledge of cryptography as it is the last piece for network security.

3.1. Lessons Learned

This project has been a journey of profound learning and growth, providing valuable insights into the realm of network security and encryption. Among the key takeaways are:

- **Comprehensive Knowledge Gain:** One of the most significant achievements of this project has been the acquisition of in-depth knowledge in the field of security. Before embarking on this endeavor, I had limited understanding of both modern encryption techniques and the intricacies of quantum-resistant encryption. The project offered a steep learning curve that allowed me to grasp the foundations and complexities of encryption, thus expanding my expertise.
- **Quantum Resistance Awareness:** The exploration of quantum-resistant encryption was particularly enlightening. This emerging field not only demands a deep dive into cryptographic concepts but also underscores the criticality of fortifying our digital infrastructure against the potential advent of

quantum computing. It has served as a reminder of the constant need to stay ahead of the curve in network security.

- Network Security Significance: In an age where the internet has become the focal point of modern life, the project underscored the paramount importance of network security. The internet's pivotal role in our daily interactions highlights the necessity of safeguarding sensitive data and communications, making security expertise increasingly essential.
- VPN Protocol Understanding: The journey into VPN protocols has provided invaluable insights into the creation and management of secure communication channels. Understanding the intricacies of VPNs empowers us to build our own security mechanisms and tailor them to meet the unique needs and expectations of the digital world.

In conclusion, this project has not only expanded my horizons in the security domain but has also reinforced the urgency of staying at the forefront of evolving security paradigms. The knowledge and skills gained are not only applicable to this project but have broader implications for ensuring the privacy and security of digital communications in an interconnected world.

3.2. Closing Remarks

This project stands as an exceptional opportunity within the CST BSc Network Security Applications Development Option. It not only reinforces the core principles taught throughout the program but also extends the learning experience, equipping students with the knowledge and skills necessary for a seamless transition into the industry. The understanding gained and the lessons learned hold immeasurable value for students specializing in this field.

4. Appendix

4.1. Approved Proposal

https://drive.google.com/file/d/136lMKuBrjESpZ38zfFUo3iF0rt35ZzeU/view?usp=share_link

4.2. Project Supervisor Approvals

5. References

- Barker, E. B., & Roginsky, A. L. (2015). *Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths*. <https://doi.org/10.6028/nist.sp.800-131ar1>
- Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2016). *Report on post-quantum cryptography*. <https://doi.org/10.6028/nist.ir.8105>
- Computer Security Division, I. T. L. (n.d.). *Post-quantum cryptography: CSRC*. CSRC. Retrieved October 26, 2022, from <https://csrc.nist.gov/projects/post-quantum-cryptography>
- Crystals*. Kyber. Retrieved October 26, 2022, from <https://pq-crystals.org/kyber/>
- IBM. What is quantum computing? Retrieved November 7, 2022, from <https://www.ibm.com/topics/quantum-computing>
- Mullvad. Experimental post-quantum safe VPN tunnels - blog. Mullvad VPN. Retrieved November 7, 2022, from <https://mullvad.net/en/blog/2022/7/11/experimental-post-quantum-safe-vpn-tunnels/>
- Open-Quantum-Safe. *Open-quantum-safe/liboqs-python: Python 3 bindings for liboqs*. GitHub. Retrieved October 26, 2022, from <https://github.com/open-quantum-safe/liboqs-python>
- Post-quantum cryptography VPN*. Microsoft Research. (2020, September 14). Retrieved October 26, 2022, from <https://www.microsoft.com/en-us/research/project/post-quantum-crypto-vpn/>
- Post-quantum cryptography*. Open Quantum Safe. Retrieved September 28, 2022, from <https://openquantumsafe.org/post-quantum-crypto.html>

6. Change Log

2023.10.20 – Version 1

- Initial Submission

2023.10.30 – Version 2: Revision after integration of post quantum algorithms with OpenVPN

- Section 2.5.2.2:
 - Changed the content of this section as the integration of post quantum algorithms and the VPN application is fully implemented after supervisor's suggestion.
- Section 2.5.3:
 - Added a table of requirements and a note.
 - Moved the configuration of OQS-provider from Section 2.5.4.3 to this section.
- Section 2.5.4.1:
 - Changed the section from the original post quantum client-server application to setup script instructions.
- Section 2.5.1:
 - Changed the details of the manual for both client and server VPN applications as it is now quantum resistant.
- Section 2.5.7:
 - Changed the software architecture diagram and explanations to show how the post quantum algorithms are implemented into the application.
- Section 2.5.8:
 - Deleted this section due to the completion of the original scope (Milestone 2).
- Section 2.6.2:
 - Changed the testing of the VPN application back to the original test cases from the proposal.
- Section 2.7/9/10/11/12:
 - Changed the content of this section as the integration of post quantum algorithms and the VPN application is fully implemented after supervisor's suggestion.