**Project Progress Report**

David Yi

The Polytechnic School, Arizona State University

IFT520: Advanced Information Systems Security

Dr. Jim Helm

November 28, 2023

**Abstract**

This project explores VPN technologies to facilitate secure communication between two different computers. To create a lightweight and open-source VPN peer to peer client, the WireGuard framework was utilized in addition to initially the Rust programming language, and ultimately the Python programming language. The client, while still bare bones, is fully functional and capable of basic file transfer through the VPN tunnel.

Keywords: VPN, WireGuard, Rust, Python

**Introduction/Overview**

**Problem Statement**

Securing data and protecting its privacy are often seen as aspects of security. They are closely connected. While organizations prioritize data security individual users determine their personal threat landscape. Take measures to keep their data safe. The Federal Trade Commission (2021) provides guidelines on how to ensure the security of user data, such, as updating software, including the operating system with the security patches. It also recommends using passwords or multifactor authentication for accounts and creating backups through cloud services or external storage devices.

While these steps are crucial additional considerations arise when sharing or uploading data online. Questions about protecting data during transit and the visibility of users’ identifiable information to actors become important. The impact of compromised data varies for organizations and individuals. For corporations compromised data at rest or in transit can result in loss and a decline, in consumer trust although the consequences may be less severe compared to individuals. Individuals face risks as compromising their data can jeopardize their entire livelihoods. Identity theft is an example that includes transactions, tax filing frauds, health insurance claims abuse and more highlighting the serious implications of personal data breaches (Luthi, 2022).

There are ways to ensure the security of data. Ultimately it is, up to the user to take responsibility, for safeguarding their data both when its stored and when its being transferred.

**Significance of Study**

The current state of data security poses a challenge due, to the use of expensive solutions that require continuous payments for comprehensive data protection. While modern computers have built in encryption for data storage there is a gap in easily accessible methods to ensure data safety during transmission. In todays interconnected society it is crucial to prioritize the security of data as it travels through networks and the internet.

Virtual Private Networks (VPNs) have emerged as a solution to this challenge by establishing connections encrypting internet traffic and protecting users online identities especially when using public networks. However one major hurdle is the reliance on paid VPN services, which can create constraints for individuals and organizations. Furthermore concerns about how VPN providers handle user data only exacerbate the issue.

This research project aims to address the pressing need for cost solutions that protect data during transit. Its main objective is to make data security accessible, to everyone regardless of their situation. By doing this project seeks to empower users with tools that enhance privacy and strengthen data security in our increasingly interconnected digital world.

By prioritizing affordability and accessibility this initiative aims to close the existing divide by offering individuals and organizations the tools they need to protect their data and address the difficulties presented by todays data security environment.

**Objective**

The aim of this project is to create a user app that caters, to individuals or organizations who want to protect their data when it moves between devices. The app will automate the setup of elements like authentication, encryption and iptables rules. Users will have the flexibility to choose where their data goes ensuring it can work across platforms while keeping it encrypted during transit. We plan to build this app using Rust, which's a programming language that’s source and designed with security measures, against buffer overflow attacks. Additionally, we're making sure the app can be easily deployed without needing permissions so anyone can use it without rights.

**Design Process**

The creation of a peer, to peer VPN client using Python and WireGuard involves considering aspects such as the architecture, user experience and system functionality.

During the stage we focus on designing the architecture of the VPN client. This includes outlining components like modules for discovering peers, secure exchange mechanisms and data encryption protocols. We also make decisions on how to integrate WireGuard, including configuring virtual network interfaces. Our aim is to create an architecture that's scalable, modular and efficient.

At the time we prioritize creating a user interface that simplifies the configuration and management of the VPN client. We take into account features that empower users, such as a setup process, real time status updates and straightforward controls for adding or removing peers. Our goal is to design an interface that not looks good but also enhances user experience.

Throughout this process security remains a priority. We focus on exchange methods, authentication protocols and robust encryption techniques. When integrating WireGuard into our design choices align with our clients security objectives to ensure data confidentiality and integrity. This phase involves making informed decisions, about algorithms, key management practices while implementing measures to prevent potential vulnerabilities.

Integrating WireGuard, into the Python application is a part of the design process. It involves setting up WireGuard generating public keys and establishing virtual network interfaces. The design focuses on ensuring compatibility between the VPN client and WireGuard prioritizing simplicity and efficiency.

To design the testing process we create test cases to verify the functionality and security of the VPN client. We examine components through unit tests while assessing system performance through integration tests. Our testing approach includes scenarios that simulate peer interactions, key exchange and secure data transmission to identify and address any issues.

We strategically approach documentation design to provide comprehensive guidance for users and developers. This includes crafting installation instructions, configuration guidelines and code comments that are easy to understand. Our documentation design ensures that users have resources for deploying and optimizing the VPN client.

The decision to make this project open source on platforms like GitHub is part of our collaboration strategy. We create an inviting repository structure with contribution guidelines to foster an inclusive community. Through this design we encourage community engagement, feedback and collaborative development for improvement over time.

Our continuous improvement design establishes a framework, for monitoring updates in Python and WireGuard.

The plan describes a way to deal with identified weaknesses releasing improvements based on user input and making sure that the VPN client adapts dynamically to meet the evolving needs of its users.

Essentially creating a peer, to peer VPN client involves an iterative process. This includes considering the architecture ensuring an user experience implementing security measures integrating with systems, devising testing strategies providing clear documentation and following collaborative development principles. Each aspect is carefully designed to contribute to the success, functionality and longevity of the VPN client.

**Requirements**

To create a solution, for safeguarding user data during transmission our implementation plan incorporates the usage of the Rust programming language GitHub for version control and the integration of the WireGuard open source VPN protocol.

The first step involves establishing a GitHub repository where we can collaborate and manage versions of our project. This repository will include a README file that provides insights into the projects purpose instructions for installation and guidelines for usage. We will prioritize maintaining documentation within this repository to promote transparency and facilitate collaboration.

Next we will leverage the Rust programming language to build the core functionality of our data protection solution. Rusts safety features, advanced memory management capabilities and performance optimization techniques will play a role in ensuring encryption and transmission of data. Our implementation efforts will focus on integrating measures that prevent programming errors aligning with Rusts safety objectives.

A significant step, in our plan is seamlessly integrating the WireGuard open source VPN protocol into our solution. This integration will provide encryption. Secure tunneling capabilities essential to enhance security during data transmission.

WireGuards simplicity and efficiency will play a role, in achieving the desired level of security and performance.

To make it easy for users to set up and activate their VPN connections we will create a user interface (UI). This UI will focus on simplicity and intuitiveness providing an experience.

We are committed to keeping project documentation on GitHub up to date. This will give users information on how to install, configure and effectively use our data protection solution. Additionally we will create user guides and tutorials to further assist users in maximizing the benefits of our tool.

By following this implementation plan, which involves using the Rust programming language utilizing GitHub for version control and integrating with the WireGuard VPN protocol our ultimate goal is to deliver an open source data protection solution. We aim to empower users by enabling them to secure their data during transmission while prioritizing privacy and security in todays interconnected world. The thoughtful combination of these technologies ensures that our solution is not effective but accessible and usable, for a diverse range of users.

**Scope**

The main goal of this project is to create a data protection system that's both open source and ensures the highest level of security and accessibility. Our primary focus is to provide users with a way to keep their data safe during transmission without any limitations. Protecting user privacy and data security is our priority as we aim to enhance the safeguarding of information, during communication and data sharing.

To accomplish this we will utilize Rust, a programming language for system development to efficiently manage low level tasks. With a focus on safety, performance and memory management our core objective involves creating a networking application in Rust that guarantees data protection.

We will use GitHub as our platform for version control allowing us to track code changes manage project related issues and promote collaboration among team members and the wider open source community. This approach ensures transparency and accessibility throughout the project encouraging contributions from others.

To enhance encryption capabilities and enable tunneling functions we will seamlessly integrate the WireGuard VPN protocol into our solution. Prioritizing simplicity, security and efficiency, in data encryption and transmission this integration will provide users with a VPN protocol that does not require any subscription charges. Our aim is to maintain a design that delivers performance while adhering to strict data security standards.

Project Deliverables;

Creating a data protection solution using the Rust programming language.

Setting up a GitHub repository, with version control and thorough documentation for transparency and ease of access.

Integrating the WireGuard VPN protocol into the solution to improve data encryption and transmission capabilities.

Designing user friendly interfaces that simplify configuration and usage of the data protection solution.

Continuously maintaining the open source project on GitHub encouraging community contributions and collaboration.

Throughout this project we will prioritize following industry practices, in software development, cybersecurity and open source collaboration. Our ultimate objective is to develop a tool that not supports but also enhances data protection during transmission providing users with an user friendly solution.

**Development Process**

The journey of creating a peer, to peer VPN client using Python and WireGuard begins with planning and initiation. In this phase we define the projects goals and scope focusing on features like discovering peers exchanging secure keys and ensuring strong data encryption. We diligently set up the development environment by installing Python and WireGuard as creating a virtual environment for efficient dependency management. To facilitate collaboration and version control we also implement a system such as Git.

Moving forward we delve into the implementation details of the VPN client. We develop mechanisms for peers to easily discover and connect with each other within the network. Secure protocols are established for exchanging keys to ensure the confidentiality and integrity of communications. By leveraging techniques we implement authentication mechanisms to enhance the security of our VPN client.

WireGuard, renowned for its efficiency and security advantages is seamlessly integrated into our Python application. This integration includes configuring WireGuard settings such as generating public keys, for each peer. Additionally we create virtual network interfaces to construct tunnels through which data can be transmitted.

An essential aspect of our development process focuses on designing and implementing a user interface. This user interface allows users to easily set up and manage the peer, to peer VPN client. It offers features like adding or removing peers, real time monitoring of status and overseeing data transfers. To ensure the reliability and security of the VPN client testing and debugging procedures are carried out. Unit tests examine components closely while integration tests ensure that the overall system functions seamlessly. Any identified issues or vulnerabilities are diligently. Resolved to strengthen the application against threats.

Documentation plays a role in the development process. We create comprehensive documentation that includes installation instructions, configuration guidelines well as informative code comments. This documentation serves as a resource for users and collaborators facilitating onboarding and ensuring a clear understanding of how the VPN client works.

Following open source principles we release our project on platforms, like GitHub to encourage community engagement, feedback and contributions. Open source collaboration allows us to benefit from expertise and insights within the community.

The development process doesn't end with the release; instead it embraces improvement as an ongoing mindset.

Keeping track of Python and WireGuard updates fixing vulnerabilities and releasing updates based on user feedback are aspects of improving and stabilizing the VPN client. This iterative method guarantees that the developed solution not fulfills its goals but also adapts to meet the changing requirements of its user community.

**Tools**

When it comes to creating a solution that ensures data security during transmission my approach revolves around using the Rust programming language GitHub, for version control and the WireGuard open source VPN protocol. Rust is known for its focus on safety, performance and memory management which makes it an ideal choice for building networking applications. The languages ability to prevent programming errors like memory leaks and buffer overflows aligns with our goal of developing a data protection solution.

For managing version control and enabling development GitHub stands out as a platform that offers features for tracking changes handling issues and promoting collaboration among team members and the wider open source community. Its popularity and accessibility guarantee transparency and availability of the project so that continuous improvement can be achieved through contributions, from a community. This collaborative approach allows us to create a documented solution that can evolve and adapt over time.

At the heart of this solution lies the integration of the WireGuard open source VPN protocol known for its simplicity, security and efficiency in encrypting data while providing tunneling capabilities. This strategic decision perfectly aligns with our objective of offering users a VPN protocol without any subscription constraints.

WireGuards focus on performance is reinforced by its efficient design ensuring that it delivers a solution without sacrificing speed.

This project aims to create an open source solution, for protecting data during transmission by leveraging the strengths of the Rust programming language GitHub, for version control and the WireGuard VPN protocol. By selecting these tools and meeting requirements we can develop a solution that not only prioritizes user privacy but also promotes data security in an accessible and efficient way.

**Technical Description of Project**

**Testing and Results**

**Summary and Conclusions**

In summary combining WireGuard and Python to create a source peer, to peer VPN client brings together innovation, security and collaborative development. By utilizing the secure features of the WireGuard VPN protocol in the Python environment this project not only ensures strong data protection during transmission but also embodies the spirit of working together on open source projects.

The thought out architectural design, user friendly interface considerations and meticulous security measures demonstrate a dedication to creating a solution that not meets technical requirements but also enhances user experience. Seamlessly integrating WireGuard into the Python environment contributes to the projects simplicity, efficiency and adaptability.

The decision to make this project open source on platforms like GitHub reflects a belief in the power of collaboration. By inviting the community to engage contribute ideas and provide feedback this project goes beyond efforts and creates a dynamic ecosystem where collective expertise refines and evolves the VPN client over time.

Moreover prioritizing improvement is evident through monitoring addressing vulnerabilities promptly and incorporating user driven updates. This ensures that the peer, to peer VPN client remains relevant and resilient as technology continues to evolve.

In terms combining WireGuard and Python to create a source peer, to peer VPN client demonstrates a dedication to accessibility, security and community driven progress. As this project begins its journey as an open source initiative it not offers users a tool to protect their data. Also becomes a collaborative effort that empowers individuals and the wider community to contribute to and benefit from a shared vision of secure communication between peers.

The integration of WireGuard into the Python application is a part of the design process. This includes configuring WireGuard settings generating public keys and establishing virtual network interfaces. The design ensures compatibility between the VPN client and WireGuard while prioritizing simplicity and efficiency.

Designing the testing process involves creating test cases to verify the functionality and security of the VPN client. Unit tests focus on components while integration tests evaluate system performance. The testing design includes scenarios that simulate peer interactions, key exchanges and secure data transmission in order to identify and address any issues.

The documentation design is carefully crafted to provide thorough guidance for users and developers alike. Installation instructions, configuration guidelines and code comments are all designed with ease of understanding, in mind.

This documentation design ensures that users have access, to resources for deploying and optimizing the VPN client.

The decision to make the project open source on platforms like GitHub is part of the collaboration plan. This involves creating a welcoming repository structure, clear guidelines for contribution and fostering an open community. The plan encourages community involvement, feedback and collaborative development to improve the project over time.

The design for enhancement establishes a framework for monitoring updates in Python and WireGuard. It outlines an approach to addressing identified vulnerabilities releasing updates based on user feedback and ensuring that the VPN client dynamically evolves to meet the changing needs of its user base.

In essence designing a peer to peer VPN client follows an iterative approach. It considers architecture, user experience, security measures, system integration, testing strategies, clarity, in documentation and principles of development. Each aspect is carefully designed to contribute to the success, functionality and longevity of the VPN client.

**References**

Donenfeld, J. A. (n.d.). *WireGuard: fast, modern, secure VPN tunnel*. https://www.wireguard.com/

Federal Trade Commission. (2021, May). *Protect your personal information and data*. Consumer Advice. https://consumer.ftc.gov/articles/protect-your-personal-information-and-data

Github. (n.d.). *Build software better, together*. GitHub. https://github.com/about

Kaspersky. (2023, July 31). *What is VPN? How It Works, Types of VPN*. www.kaspersky.com. https://www.kaspersky.com/resource-center/definitions/what-is-a-vpn

Luthi, B. (2022). What is identity theft and how do I make sure it doesn’t happen to me? *www.experian.com*. https://www.experian.com/blogs/ask-experian/what-is-identity-theft/

*Rust Programming Language*. (n.d.). https://www.rust-lang.org/

**Appendix**