Virual Private Network (VPN) Simulation

Design Document by Group 16
Version 1.0
October 8, 2024

Anvit Gupta
22114009
anvit_g@cs.iitr.ac.in

Ayush Ranjan
22114018
ayush_r@cs.iitr.ac.in

Indranil Das
22114037
indranil_d@cs.iitr.ac.in

Sarvasva Gupta
22114086
sarvasva_g@cs.iitr.ac.in

Souvik Karmakar
22114096
souvik_k@cs.iitr.ac.in

Vineet Kumar
22114107
vineet_k@cs.iitr.ac.in

1. Summary of Project

This mini project involves simulating a Virtual Private Network (VPN) to ensure secure and encrypted communication between clients and a server. The VPN environment will be set up using OpenVPN or equivalent software. The project will also include performance analysis by measuring the impact of VPN usage on network latency and throughput. The main focus is on securing communication (data confidentiality) and evaluating the performance overhead due to encryption and tunneling mechanisms.

2. High Level Design

2.1. Structured Analysis - Data Flow Diagrams (DFDs)

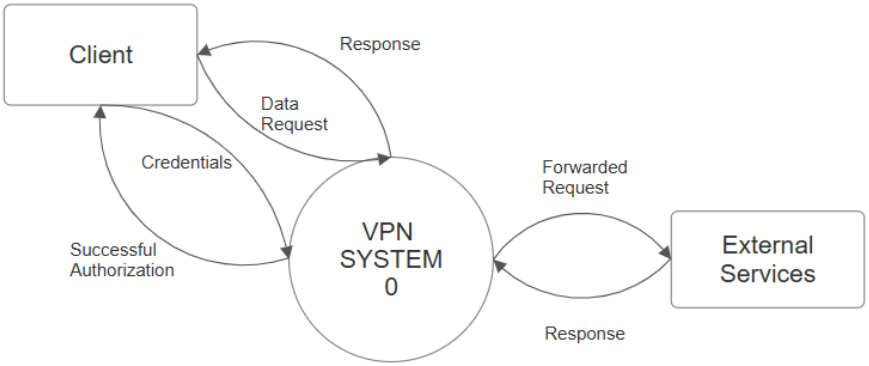

Figure 1: Context Diagram

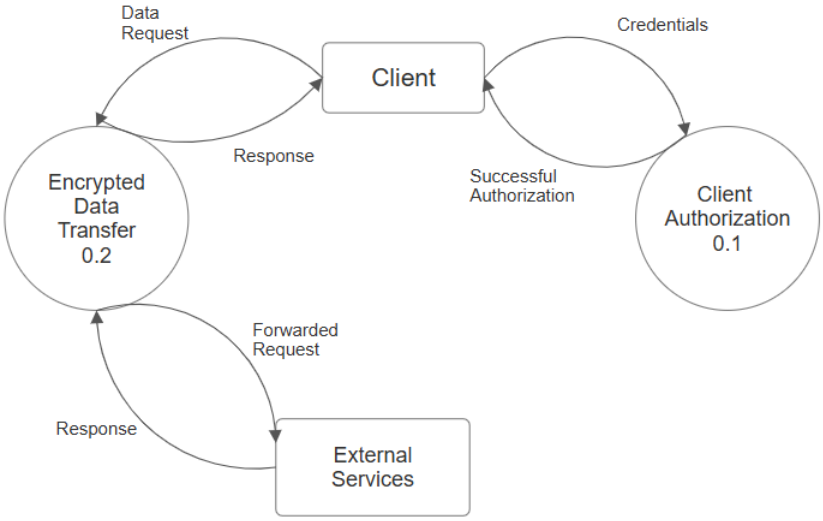

Figure 2: Level 1 DFD

2.2. Structured Design

1. Client Authorization Module

- **Purpose:** Verifies client credentials and returns authorization status.
- **Function:** Receives client credentials and returns a boolean (`true` or `false`) indicating whether the client is authorized.
- **Output:** Sends the boolean (`true` or `false`) to the Encrypted Data Transfer Module.

2. Encrypted Data Transfer Module

- **Purpose:** Secures data transfer based on the authorization status received from the Client Authorization Module.
- **Function:** If the authorization status is `true`, initiates encrypted data transfer. If `false`, blocks data transfer.

3. Low Level Design

1. Client Authorization Module

- **Data Structures:**
  - `struct ClientCredentials { string username; string password; }`
  - `bool authorize(ClientCredentials creds)`
- **Algorithms:**
  - Compare `creds` with stored values.
  - If valid, return `true`; otherwise, return `false`.

2. Encrypted Data Transfer Module

- **Data Structures:**
  - `struct DataPacket { string data; bool isEncrypted; }`
- **Algorithms:**
  - Accept the authorization result (`true/false`).
  - If `true`, encrypt and transfer data using an encryption algorithm (e.g., AES).
  - If `false`, block the transfer.

4. Applications of the Project

- **Educational Tool:** Serves as a practical learning resource for students and professionals to understand VPN technology, encryption protocols, and network security concepts.
- **Network Security Testing:** Allows for the testing and evaluation of VPN performance, encryption methods, and client-server communication under simulated conditions.
- **Prototype for Further Development:** Provides a foundation for developing more complex VPN solutions or enhancements, such as adding additional security features or support for multiple clients.