PERSONAL FIREWALL

**1. Introduction**

In today's connected world, our computers are constantly sending and receiving information over the internet. Just like you have a lock on your door to keep unwanted visitors out of your home, your computer needs protection to control who or what can access its network. This protection is called a **firewall**.

This project is about building a simple, custom firewall using Python. Our goal was to create a "security guard" for a computer's internet connection. This security guard would understand our rules about what traffic to allow or block, keep a record of everything, and let us see what it's doing. It's a hands-on way to understand how internet security works.

**2. Abstract**

This report details a **Python-based personal firewall** designed to filter internet traffic. It uses custom rules to **monitor data with Scapy** and **enforce blocks/allows via UFW**. All firewall actions are logged, and it provides both **Command Line (CLI)** and **Graphical (GUI)** interfaces for control and live monitoring. This project showcases practical skills in network security and user-friendly software development.

**3. Tools Used**

Building this personal firewall relied on several key tools:

* **Python 3:** The core programming language.
* **Scapy:** For capturing and understanding network traffic.
* **UFW (Uncomplicated Firewall):** To enforce traffic rules at the system level.
* **Tkinter:** For the graphical user interface (GUI).
* **Subprocess Module:** To allow Python to communicate with UFW.
* **Threading Module:** To keep the firewall responsive while performing multiple tasks.
* **JSON:** For defining and reading firewall rules.
* **Logging Module:** For keeping a detailed record of firewall activity.

**4. Steps Involved in Building the Project**

Building the firewall followed a clear step-by-step process:

* Setting Up the Workspace: Installed necessary tools (Python, Scapy, UFW) and organized project folders.
* Defining the Rules: Wrote down firewall rules (allow/block, IPs, ports, protocols) in a configuration file.
* Rule Engine: Programmed the logic to load rules and match them against network traffic.
* Packet Sniffer: Developed a component to capture and observe all incoming and outgoing network data.
* Firewall Core: The central brain; it configures the system's UFW firewall, processes captured traffic, logs decisions, and manages the firewall's overall operation.
* User Interfaces (CLI & GUI): Created both a text-based (Command Line Interface) and a visual (Graphical User Interface) way to control and monitor the firewall.
* Main Entry Point: The primary script that launches the entire firewall, initiating all components.

**5. Conclusion**

This project successfully built a functioning personal firewall using Python. We learned how to define security rules, how to "see" internet traffic with tools like Scapy, and how to tell the computer's own firewall (UFW) to enforce these rules. By creating both a simple command-line tool and a visual application, we made it easy to use and monitor. This project was a great way to understand the basics of protecting a computer from unwanted internet access and how different software pieces work together to achieve a security goal.

**--Karthikeya Nanduri**