# 📄 Project Report: Personal Firewall using Python

---

## 1. Introduction

The internet is full of malicious activities that can harm users and systems. Firewalls are essential components of network security that filter unwanted traffic. This project aims to create a personal, lightweight firewall using Python and Scapy that allows users to define custom rules for blocking or allowing IPs, ports, and protocols.

---

## 2. Abstract

This project involves building a CLI-based personal firewall that uses Python and Scapy to monitor real-time network traffic and enforce filtering rules. Users can specify rules to block certain IP addresses, disallow specific TCP/UDP ports, and restrict protocols. Suspicious packets are logged for auditing. The firewall acts as a learning tool for understanding packet filtering at a low level and can be expanded with a GUI or Linux iptables integration.

---

## 3. Tools Used

- **Python 3.x** – Core programming language
- **Scapy** – For packet sniffing and analysis
- **Logging module –** To log blocked and allowed packets

---

## 4. Steps Involved in Building the Project

1. **Setup and Installation**
   - Installed Python and Scapy using pip install scapy.
   - Created a virtual environment using PyCharm.
2. **Packet Sniffing**
   - Used Scapy's sniff() function to capture live network packets.
3. **Rule Definition**
   - Defined block/allow rules in a dictionary format for IPs, ports, and protocols.
4. **Filtering Logic**
   - Applied logic to drop or allow packets based on rules.
   - Used IP and TCP layers to inspect source IP and destination port.
5. **Logging System**
   - Implemented logging using Python's logging module.

    - Logged details of every blocked or allowed packet into a log file.

6. **Testing and Execution**
    - Ran the script and observed packet behavior in the console.
    - Verified logging in firewall_log.txt.

---

## 5. Conclusion

This project helped understand the basics of network packet filtering and Python-based firewall implementation. The firewall successfully blocked or allowed traffic based on user-defined rules and logged activities for further analysis. This can be further extended to include advanced rule sets, a GUI, or integration with system firewalls like iptables.