Personal Firewall Using Python – Project Report

Introduction

This project focuses on building a lightweight Personal Firewall using Python. The firewall filters incoming and outgoing network packets based on predefined rules and helps users monitor and control network traffic. It also provides basic packet logging for auditing and security analysis.

Abstract

The Personal Firewall is implemented using Python and Scapy to sniff, analyze, and filter network packets. The system applies rule-based filtering to allow or block packets depending on factors such as IP address, port number, protocol, and direction (incoming/outgoing). Suspicious or blocked packets are logged for further investigation. Optionally, the system can integrate with iptables on Linux for system-level enforcement. This project demonstrates the fundamentals of packet inspection, network rule enforcement, and personal firewall design.

Tools Used

- Python 3
- Scapy (packet sniffing & manipulation)
- Tkinter (optional GUI for monitoring)
- iptables (optional system-level rule enforcement on Linux)
- JSON for storing rules (example provided in your rule set)
- Virtual Environment (venv)
- Logging module for activity and alert logs

Steps Involved in Building the Project

1. Setting Up the Environment

- Installed Scapy and configured Python environment.
- Created JSON-based rule file containing allow/block conditions (IP, port, protocol, direction).
- Defined folder structure for firewall code, logs, and rule sets.

2. Implementing Packet Sniffing

- Used Scapy's sniff() function to capture incoming and outgoing packets.
- Extracted relevant fields: source IP, destination IP, protocol, ports, etc.

3. Rule Matching & Filtering

- Wrote a rule-matching function that checks each packet against the rule list.
- Supported attributes: direction (in/out/both), protocol (TCP/UDP/ANY), ports, IP addresses
- Example rules (from your project): Block all incoming traffic, Allow outgoing HTTP (port 80), Allow outgoing HTTPS (port 443), Allow DNS on port 53 (UDP)

4. Blocking / Allowing Packets

- For packets that violate rules → marked as *blocked*.
- Allowed packets logged or forwarded.
- Integrated optional iptables execution for real packet dropping (Linux).

5. Logging System

- Implemented logging of: Blocked packets, Suspicious traffic, Allowed packets (optional)
- Logs stored in a timestamped file for audits.

6. Optional GUI (Tkinter)

- Built a simple dashboard to: View live packet traffic, Enable/disable firewall, Load and modify rule sets, View logs in real time

7. Testing the Firewall

- Tested rules using: Ping (ICMP), Browsing (HTTP/HTTPS), DNS lookups
- Verified that:
    - Incoming packets were blocked
    - Only allowed ports (80/443/53) passed
    - Logs captured all blocked packets

Conclusion

The Personal Firewall project successfully demonstrates how packet sniffing, analysis, and filtering can be implemented using Python.
It provides a rule-based approach for controlling network traffic and logging suspicious activities.
The project enhances understanding of network protocols, packet structures, and security rule enforcement.
Future improvements may include deep packet inspection, multi-threaded sniffing, IP reputation checking, real-time alerts, and full GUI-based management.