Rahi's Personal Firewall using Python – Project Report

Introduction

With increasing concerns over network security, personal firewalls play a crucial role in monitoring and filtering network traffic. This project focuses on creating a lightweight personal firewall using Python that inspects incoming and outgoing packets, enforces custom rule sets, and optionally uses Linux's iptables for deeper integration. It aims to provide users with real-time control over their network traffic.

Abstract

The goal of this project is to develop a Python-based personal firewall that filters traffic based on predefined rules. Using scapy, it can sniff packets and analyze them in real-time. Custom rule sets allow blocking or allowing IPs, ports, and protocols. Logs are generated for auditing, and a simple GUI (built using Tkinter) enables users to monitor traffic and manage rules interactively. This firewall is suitable for basic protection and educational purposes, offering insight into how packet inspection and traffic control work.

Tools Used

- **Python 3** – Main programming language
- **Scapy** – For sniffing and analyzing network packets
- **iptables** – Optional system-level rule enforcement (Linux only)
- **Tkinter** – For optional GUI interface
- **Logging** – For packet and rule event logging

Steps Involved in Building the Project

1. **Packet Sniffing with Scapy:**
   Set up scapy to monitor both incoming and outgoing traffic. Applied filters to capture packets based on protocols (TCP, UDP, ICMP).
2. **Define Rule Engine:**
   Created a rule engine that can block or allow traffic based on:
   - Source/Destination IP

- Source/Destination Port
- Protocol (TCP/UDP/ICMP)

3. **Rule Application:**
   Each packet is compared against user-defined rules. If matched, the action is taken (allow or drop). Optionally, iptables commands can be triggered to apply rules at the system level.
4. **Logging:**
   Suspicious or dropped packets are logged with timestamp, IP info, port, and protocol to an external log file for later auditing.
5. **Graphical User Interface (Optional):**
   Using Tkinter, a simple GUI was implemented to:
   - Display live traffic stats
   - Add/remove firewall rules
   - View logs in real-time
6. **Testing and Validation:**
   Simulated various packet types and used network tools to test blocking/allowing behavior. Verified GUI control and logging outputs.

Conclusion

The personal firewall built in this project provides fundamental traffic inspection and control features. While it may not replace enterprise-level solutions, it serves as an excellent educational tool to understand how packet filtering, rule matching, and network logging work. Future enhancements could include stateful inspection, NAT translation, automatic rule updates, and enhanced user interface with live charts or threat intelligence integration.