Personal Firewall using Python

Introduction:

With the ever-growing reliance on internet-connected systems, protecting computers from malicious traffic is critical. A personal firewall is a local security system that monitors and filters incoming and outgoing network traffic based on predefined security rules. This project aims to build a lightweight personal firewall using Python, capable of sniffing packets, applying custom filtering rules, logging suspicious activity, and optionally enforcing rules using system-level controls.

Abstract:

This implements a personal firewall that inspects real-time network packets using Scapy and filters them based on user-defined rules (like blocking specific IPs, ports, or protocols). It also supports logging blocked traffic for auditing purposes. For enhanced security, it can use iptables (on Linux) to enforce system-level rules. An optional GUI is provided using Tkinter to allow users to monitor traffic logs in real-time. This project serves both as an educational tool and a practical security component for personal computers.

Tools Used:

Python

Scapy

Iptables

Tkinter

JSON

Steps Involved in Building the Project

- **Environment Setup**
  Install Python, Scapy, and optionally Tkinter using pip and apt. Create the required project folder and files.
- **Define Rules in JSON**
  Users create a rules.json file to specify which IPs or ports should be blocked or allowed.
- **Sniff Network Packets (Scapy)**
  Using Scapy's sniff() function, incoming and outgoing packets are captured and passed to a callback function for inspection.

- **Match Packets Against Rules**
  Each packet is compared with rules. If a rule matches and the action is block, the packet is logged and considered rejected.
- **Log Suspicious Activity**
  Blocked packets are logged to firewall_log.txt with timestamps using a custom logging module.
- **Apply iptables Rules**
  If enabled via CLI flag, the script uses subprocess to apply matching iptables rules, providing actual blocking at the OS level.
- **GUI for Monitoring**
  A Tkinter-based GUI displays log entries in real time and provides basic control over firewall operation.

Conclusion

This demonstrates how Python can be used to build a basic yet functional personal firewall. It combines real-time packet sniffing, rule-based filtering, and optional system-level enforcement to protect a Linux system from unwanted traffic. While it is not a substitute for enterprise-grade solutions, it provides a hands-on understanding of packet inspection and local security filtering — valuable for cybersecurity education and experimentation.