Internship Project Report

Title: Personal Firewall using Python (CLI-based)
Name: SNEHASHREE N
Internship Organization: Elevate Labs
Submission Month: July 2025

1. Introduction

A firewall is a fundamental component in cybersecurity, acting as a barrier between trusted and untrusted networks. The aim of this project was to create a lightweight personal firewall using Python. This firewall allows traffic filtering based on user-defined rules, logs suspicious activity, and demonstrates packet inspection in real time.

2. Objective

To build a CLI-based personal firewall using Python that can:

- Inspect and analyze network traffic using Scapy
- Block or allow packets based on predefined rules (IPs, ports, protocols)
- Log blocked packets to a file for auditing
- Demonstrate basic defensive security mechanisms

3. Tools and Technologies

- **Programming Language:** Python 3
- **Libraries Used:**
  - Scapy (packet sniffing and parsing)
  - Logging (built-in Python module)
- **Platform:** Parrot OS (Linux)
- **Attacker Environment:** Kali Linux

4. Methodology

1. **Interface Configuration**:
   - Verified correct interfaces using ip a

   - Used ens33 as the primary network interface for sniffing

2. **Rule Definition**:
   - Block incoming traffic from specific IPs (e.g., Kali attacker)
   - Block selected ports (22 for SSH, 80 for HTTP)
   - Block ICMP (to block ping attacks)
3. **Packet Capture & Filtering**:
   - Used Scapy's sniff() to capture live packets
   - Filtered packets using rules defined in a dictionary
   - Logged results in terminal and firewall.log
4. **Testing with Attacker Machine**:
   - Ran ping and nmap from Kali
   - Verified logs and terminal outputs on Parrot

5. Project Output

- CLI-based firewall that starts with:
- sudo python3 firewall_cli.py
- Real-time packet log output (ALLOWED/BLOCKED)
- Permanent logging in firewall.log

Example Log Output:

```
[BLOCKED] IP / ICMP 10.10.1.17 > 10.10.1.13 => Blocked IP 10.10.1.17

[ALLOWED] IP / TCP 10.10.1.10:51234 > 10.10.1.13:443
```

6. GUI (Attempted Feature)

A GUI version was also developed using Tkinter. It provided a window to view logs in real time. However, due to threading limitations and capture issues in GUI context, the final working version submitted is CLI-based.

7. Screenshots

Screenshots attached in the screenshots/ folder:

- Firewall running (CLI)
- Attacker sending pings and scans
- Interface IP confirmation
- Log file contents
- GUI window (optional feature)

8. Conclusion

This project helped reinforce core concepts in:

- Network packet inspection
- Python threading and real-time logging
- Defensive security implementations

The CLI-based personal firewall is lightweight, effective, and can be extended for more complex rule sets or a GUI in the future.

9. Future Scope

- Full-featured GUI dashboard
- Dynamic rule management
- Integration with iptables for enforcement
- Notification system for critical alerts

Submitted by:
SNEHASHREE N
Elevate Labs Internship
July 2025