ELEVATE LABS – FINAL PROJECT REPORT
Intern Name: Ved Chordia
Internship Domain: Cybersecurity
Submission Date: 27/07/2025
Project Guide:

Final Project Report

Project Title(s):

1. Web Application Vulnerability Scanner
2. Personal Firewall using Packet Filtering

Project 1: Web Application Vulnerability Scanner

Objective:

To build a Python-based security scanner that identifies basic web vulnerabilities—such as Cross-Site Scripting (XSS) and SQL Injection—by analyzing input forms and injecting common payloads.

Tools & Technologies Used:

- **Programming Language:** Python
- **Libraries:** `requests`, `BeautifulSoup`, `urllib.parse`
- **Testing Environment:** http://testphp.vulnweb.com
- **Platform:** Windows 11
- **IDE:** VS Code

Project Description:

The scanner fetches the HTML content of a given website and searches for forms or query parameters. For each form discovered:

- The tool submits test payloads to detect improper input sanitization.
- Payloads used:
    - `<script>alert(1)</script>` for detecting **XSS**
    - `' OR '1'='1` for detecting **SQL Injection**

If the website reflects the payload back or fails to handle it securely, the scanner flags it as a potential vulnerability.

Key Features:

- Automated form detection and payload injection
- Prints clear success/failure results in the terminal
- Uses modular and readable Python code for easy extension

Project 2: Personal Firewall using Packet Filtering

Objective:

To develop a lightweight Python-based personal firewall that monitors all network packets and filters them based on user-defined rules to allow or block traffic accordingly.

Tools & Technologies Used:

- **Programming Language:** Python
- **Library:** Scapy
- **Config File:** `rules.json`
- **Logging:** `firewall_log.txt`
- **Platform:** Windows 11
- **Editor:** VS Code

Project Description:

This project involves real-time packet sniffing and rule-based filtering. The firewall reads user-specified rules from `rules.json`, which define:

- Protocol (TCP, UDP, ICMP)
- Port (if applicable)
- Action (`block` or `allow`)

The program uses Scapy to sniff packets and compares each one to the rules. Packets are either:

- **Blocked and logged** (with reason)
- **Allowed** (with console info)
- **Unmatched packets** are passed with a note

Learning Outcomes:

- Gained practical experience with **network security** and **penetration testing**
- Learned how to use **Scapy** to sniff and analyze packets in real-time
- Understood and applied common web vulnerabilities: XSS and SQL Injection
- Developed logic for **firewall rule evaluation and logging**
- Practiced Python automation for cybersecurity use cases

Conclusion:

These projects gave me hands-on exposure to both **offensive and defensive** aspects of cybersecurity. I understood how web applications can be exploited and how network-based attacks can be prevented using simple tools. Both tools can serve as a base for more advanced projects in penetration testing and network defense.