Designing and Implementing an Educational Platform for SQL Injection Awareness and Defense

Technical Report

Presented to the Faculty of the

School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Michael Park

November 30th, 2024

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISOR

Nada Basit

1. Introduction

This technical report documents the development of a web-based educational platform designed to teach users how SQL Injection (SQLi) vulnerabilities operate and how to defend against them through experiential learning. Despite widespread awareness of SQLi as one of the most critical and common web application security flaws, it remains a persistent threat in modern systems due to improper input validation, insufficient secure coding practices, and gaps in real-world cybersecurity training. This project responds to these concerns by building an immersive, technically rigorous environment where learners progress through a series of escalating SQLi challenges, explore global cyber threat patterns through data visualizations, and complete a project-based tutorial culminating in a functioning anomaly detection system. The platform emphasizes both the theoretical underpinnings and the practical applications of SQLi defense, making it a valuable tool for students, educators, and professionals seeking to develop secure coding skills in a realistic and reflective context.

2. Project Overview and Objectives

The project had several primary objectives: to provide an interactive set of SQLi challenge modules; to develop contextual awareness through global cybersecurity data visualizations; to offer a comprehensive tutorial for building an SQLi detector; and to implement a complete, working demo of that detector system. Each of these components contributes to a holistic educational experience that builds technical skill, contextual understanding, and practical confidence. The core aim is to bridge the gap between abstract knowledge and hands-on capability in secure web development.

3. SQL Injection Challenge Modules

The platform's foundation is a sequence of twelve escalating SQLi modules that simulate real-world vulnerabilities. The modules begin with simple injection examples, such as bypassing authentication using tautologies, and progress to more advanced scenarios including union-based attacks, blind SQLi, and complex payload manipulation. The final module introduces secure coding techniques, including the

use of parameterized queries and input validation, which users must implement to defend the same applications they previously compromised. These modules were developed using HTML, CSS, JavaScript, PHP, and MariaDB, and are hosted on a Linux virtual machine using an Apache web server. Particular attention was paid to ensuring each module could demonstrate both vulnerability and resolution in a contained, risk-free environment. The author led the technical development and refinement of these modules, ensuring that they could accommodate a range of learning levels—from beginner to advanced—while still illustrating key concepts with clarity.

4. Global Cybersecurity Data Visualization

To place SQLi in the broader landscape of global cyber threats, the platform includes a dynamic data visualization dashboard. This module displays key statistics and trends such as the top countries experiencing cyber attacks, the most prevalent attack types by year, and frequency patterns specific to SQLi incidents. The visualizations are generated using Chart.js and updated based on user-selected parameters, making the module interactive and responsive. These graphs are intended not only to inform but to reinforce the real-world relevance of the technical skills users are acquiring. By tying individual coding behaviors to global cybersecurity outcomes, the module encourages a more reflective and systems-oriented perspective. The author implemented the full visualization system, including the interface design, filtering logic, and dataset integration.

5. SQL Injection Anomaly Detector Tutorial and Demo

To extend the learning experience from simulation to application, a full project-based tutorial was developed guiding users through the construction of an SQL Injection Anomaly Detector. This component introduces learners to the core principles of backend form handling, login systems, and input validation, and then walks them through integrating detection logic that identifies suspicious SQL-related input patterns. The completed demo includes a functioning login and registration system, anomaly logging features, and alert displays that inform users when a possible SQLi attempt is detected. All logic is

implemented in PHP with MariaDB as the backend. This system gives learners a practical understanding of how to apply secure coding principles in live applications, and serves as a capstone experience synthesizing earlier lessons. The tutorial and demo were fully developed by the author, including both functional implementation and inline documentation designed to support future improvements and instructional use.

6. Development Challenges and Future Enhancements

Several challenges arose during development. One of the primary technical issues was ensuring the vulnerable behaviors in the modules remained safe for exploration but did not compromise the virtual host environment. Pedagogically, it was also important to balance clarity with complexity: modules needed to be understandable yet not oversimplified. On the visualization side, performance and rendering consistency across devices required careful tuning. Looking ahead, future enhancements could include machine learning-based detection modules, expansion into other types of web vulnerabilities (such as XSS or CSRF), and learner dashboards that track progress and issue completion certificates. These improvements would continue to expand the platform's impact and scalability across educational settings.

7. Conclusion

This project successfully delivered an integrated cybersecurity education platform focused on SQL Injection, with a special emphasis on hands-on experience, real-world relevance, and pedagogical accessibility. The author's core contributions—including the SQLi module development, global threat visualization interface, the SQL Injection Anomaly Detector tutorial, and the completed demo—form the backbone of the platform's technical and educational value. The project advances the goal of cultivating a "security-first" mindset among future developers by equipping users not only with technical skills but also the ability to think critically about software safety, usability, and social responsibility. As SQLi vulnerabilities continue to endanger digital systems worldwide, platforms like this can play a crucial role in transforming how we teach and implement secure development practices.