Thesis Project Portfolio

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

(Technical Report)

Securing the Stack: Bridging SQL Injection Defense and Ethical Responsibility in Cybersecurity Education

(STS Research Paper)

An Undergraduate Thesis

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Michael Park

Spring, 2025

Department of Computer Science

Table of Contents

Sociotechnical Synthesis
(Executive Summary) (Level II)

Ethical Horizons in Cybersecurity Education: Balancing Technical Skills with Social Responsibility

"Security is not a product, but a process." — Bruce Schneier

Securing web applications against vulnerabilities remains a pressing challenge in cybersecurity engineering, particularly as threats such as SQL injection (SQLi) persist despite decades of documented preventive measures. Throughout my senior thesis, I pursued two interconnected projects to address this issue: a technical project developing an experiential SQLi defense training platform, and an STS research project examining how cybersecurity education can better integrate ethical responsibility alongside technical instruction. Though distinct, these projects are unified by a common motivation: the realization that technical knowledge alone is insufficient to prevent security failures if broader organizational and cultural factors are ignored. Together, they demonstrate that sustainable cybersecurity practice requires both technical fluency and sociotechnical awareness. STS perspectives reveal how engineering solutions must align not only with functional needs but also with ethical obligations to society, especially in fields like cybersecurity where the consequences of failure are profound.

The technical portion of my thesis produced an interactive educational platform designed to teach users how SQL injection vulnerabilities arise and how to defend against them. Developed using HTML, JavaScript, PHP, and MariaDB, and hosted on a Linux virtual machine, the platform consists of twelve escalating modules simulating real-world SQLi attack and defense scenarios. Users first exploit vulnerabilities, then patch them using best practices like input validation and parameterized queries. A key feature of the platform is its emphasis on reflective learning: users are exposed to cases where institutional pressures, such as rushed development timelines, led to critical security oversights. By simulating both technical failures and their organizational causes, the platform builds not only technical

competence but strategic, context-aware thinking. The potential significance of this work lies in equipping future developers to recognize vulnerabilities early, advocate for security in organizational contexts, and cultivate a "security-first" mindset critical to defending real-world systems.

In my STS research, I analyzed how cybersecurity education often emphasizes technical mastery while failing to adequately prepare students for the ethical and institutional challenges they will encounter. Drawing on theories of sociotechnical systems and institutional momentum from scholars like Hirsh and Sovacool, I argued that merely teaching students how to fix vulnerabilities like SQLi is insufficient unless they also understand why these vulnerabilities persist. My research revealed that developers often know how to code securely, but organizational pressures—deadlines, cost-cutting, inadequate ethical reflection—lead them to deprioritize security. I proposed an experiential learning model that integrates technical simulations with ethical deliberation and case study analysis, encouraging students to reflect not only on "how" to secure systems but "why" secure practices are often neglected in the field. The anticipated deliverable is an educational module that cultivates both technical ability and moral responsibility in cybersecurity learners.

Reflecting on these projects through STS frameworks deepened my understanding of engineering ethics. By examining technical, organizational, and cultural elements together, it became clear that secure software development cannot be reduced to technical procedures alone. Moral frameworks such as virtue ethics, deontology, and utilitarianism illuminate the stakes: a utilitarian perspective emphasizes minimizing societal harm through better defenses; a deontological view stresses developers' duty to uphold best practices regardless of pressures; virtue ethics focuses on cultivating habits of diligence, prudence, and integrity among engineers. Even moral relativism reminds us that differing organizational cultures shape how security is valued and practiced. Integrating STS perspectives strengthens ethical responsibility by making explicit the broader consequences of technical decisions and empowering engineers to resist institutional pressures that compromise safety.

I would like to acknowledge the support of my technical team members, Jared Conway, Lilli Hrncir, and Sami Kedir, whose collaboration enriched the technical development of the project. I am especially grateful to my advisors, Professor William Davis and Professor Nada Basit, for their insightful guidance and encouragement throughout this process. Finally, I am thankful for the University of Virginia's School of Engineering and Applied Science for providing the resources and academic environment that enabled this work.