# COSC 439: Operating Systems Project

## Title: USB HID Keystroke Injection: Attack Vector and OS-Level Defense Mechanisms

## Objective

The objective of this project is to explore USB Human Interface Device (HID) security from both offensive and defensive perspectives, demonstrating core operating system concepts including kernel module development, device driver interaction, process management, and system security. The user will implement a USB HID keystroke injection device using Arduino firmware, develop a Linux kernel module to monitor USB devices, and create a user-space daemon to analyze input events and detect suspicious keystroke patterns. This project will demonstrate the complete lifecycle of OS security: understanding vulnerabilities at the hardware-software interface, implementing kernel-level monitoring, managing user-space processes, and analyzing system behavior under attack.

## Features to Implement

**Part 1: USB HID Attack Vector (Arduino Implementation)**
Implement firmware for an Arduino Micro that enumerates as a USB keyboard and executes a pre-programmed keystroke sequence. This demonstrates how USB HID devices interact with the OS without requiring custom device drivers.

**Part 2: Linux Kernel Module for USB Device Detection**
Develop a (LKM) loadable kernel module that monitors USB bus activity and logs HID device connections. The module will register USB notifiers, identify HID class devices, extract device information, and create /sys entries to expose device lists to user space.

**Part 3: User-Space Input Monitoring Daemon**
Create a user-space daemon that monitors input events from /dev/input/ devices and analyzes keystroke patterns to detect potential injection attacks. Implement algorithms for abnormally fast keystroke rates, consistent inter-keystroke timing, and rapid modifier key sequences. Using a **weighted scoring system**, detect potential injection attacks.

**Part 4: Integration and Defense Analysis**
Demonstrate the complete system working together and analyze detection accuracy, timing windows, false positive rates, evasion techniques, and performance impact.

**Weighted Scoring System**
**IKT** = Inter-Keystroke Timing

**Malicious Score** = **100**

| Observation | Signature | Score |
|---|---|---|
| **Keystroke Timing** (Speed & Consistency) | Input characterized by consistently **Low IKT** (very fast) and **Low Jitter** (perfect consistency/no variation). | +25 (Moderate) |
| **Behavior/Intent** (Payload Execution) | Execution of **rapid system commands** designed to run a malicious payload (e.g., non-text, complex input). | +75 (Critical) |
| **Human Input Signature** | **Backspace** command or a **Pause > than 1 second**. | -100 |

## Requirements

1. **Progress Report:** Submit a progress report outlining encountered challenges, how you have solved them, the current status, and forthcoming steps. Upon submission, feedback will be given for project adjustment based on the provided feedback. **(1 pt)**
2. **Code Implementation:** Develop working implementations of Arduino firmware **(2 pts)**, kernel module **(2 pts)**, and user-space daemon **(2 pts)**. **(6 pts)**
3. **Technical Report:** Prepare a detailed report consisting of the following sections **(5 pts)**:
    - **Introduction to the Project: Purpose, Overview, and Scope**
      Define the project's objectives and its significance within OS security. Introduce USB HID security concepts and highlight the scope of the project.
    - **Attack Implementation: Arduino USB Enumeration and Payload Design**
      Describe the implementation of the Arduino USB HID attack vector.
    - **Defense Mechanisms: Kernel Module and Daemon Architecture**
      Present the design and implementation of kernel module and user-space daemon for detection.
    - **OS Concepts Analysis: Technical Aspects and Development Insights**
      Discuss kernel programming, device drivers, process management, and system security mechanisms.
    - **Testing and Results: Analysis of Detection Effectiveness**
      Showcase detection accuracy, timing analysis, and performance evaluation.
    - **Security Analysis: Vulnerabilities and Defense Effectiveness**
      Analyze USB HID vulnerabilities and effectiveness of OS-level defenses.
    - **Conclusion: Insights and Significance of OS Security**
      Summarize key findings and lessons learned about OS security.
4. **Presentation:** In person presentation that focuses on the technical aspects of the project. Utilize slides to highlight project goals, algorithms employed, implementation details, evaluations, challenges encountered, and insights gained. Additionally, ensure the presentation includes a live demonstration of the project to provide a practical illustration of its functionality. **(5 pts)**
5. **Retrospective and Contribution Report:** Reflect on the Operating Systems (OS) course, summarizing significant lessons learned, their practical relevance, and their impact on understanding OS principles. **(1 pt)**

## Deadlines

- **Progress Report:** November 17, 2025
- **Presentation and Demo:** December 3–8, 2025 (In person)
    - Presentations will take place in the professor's office (YR 456) or in the library (YR 454).
    - The professor may ask questions or request modifications to the project or source code to verify that the work was done by the students and not generated by GenAI or copied from online sources.
    - Time slots will be provided via Calendly, and students should book a slot according to their convenience.
- **Source Code Submission:** December 8, 2025
- **Technical Report:** December 11, 2025
- **Retrospective and Contribution Report:** December 11, 2025