SYSC 4907 – Fall 2026/Winter 2027

Supervisor	Jason Jaskolka
Co-supervisor	
Course section	SYSC 4907 H
Project ID	H2
Project title	Threat Pilot: A Next-Generation Threat Modeling Platform
Project description	Are you passionate about building secure software systems and shaping the future of cybersecurity? Join us in developing *Threat Pilot*, a next-generation threat modeling platform that empowers software engineers to systematically identify, analyze, and mitigate security risks throughout the entire software development lifecycle (SDLC). Threat modeling is a cornerstone of secure software engineering. It enables teams to systematically uncover potential attack scenarios, vulnerabilities, and risks early in the development process so that security can be designed in from the start and not bolted on at the end. By modeling threats, system defenders gain insight into attackers' motivations, likely attack vectors, and critical assets, driving smarter security decisions and more robust system designs. Current tools, such as OWASP Threat Dragon, OWASP pytm, Microsoft Threat Modeling Tool, Mozilla SeaSponge, Cairis, Irius Risk, Threat Composer, and TaaC-AI, offer valuable features but also have notable limitations in usability, integration, and requirements management. This *multi-year project* aims to create a new, open, and extensible threat modeling platform (known as *Threat Pilot*) that learns from the strengths of existing tools while addressing their weaknesses. The long-term vision is a tool that seamlessly integrates into the SDLC, supporting threat analysis, security requirements elicitation and traceability, and evidence-based security assurance, all within a collaborative and user-friendly environment. **Project Objectives:** Phase 1 of this multi-year project aims to: 1. Conduct a comparative analysis of existing threat modeling tools to identify strengths, weaknesses, and integration gaps that can inform Threat Pilot's requirements and design priorities. 2. Design and prototype the core architecture of Threat Pilot, focussing on creating a modular, extensible architecture with clearly defined components (e.g., system model editor, threat identification engine, threat visualization, requirements traceability module, reporting module, etc.). 3. Conduct experiments to validate critical design decisions to ensure the platform is both powerful and practical 4. Build a foundation for future teams by delivering clear documentation, a well-structured codebase, and a roadmap for ongoing development.

	Project Expectations: This project is open to students enrolled in the Software Engineering program. 1. **Application of Software Engineering Principles:** You are required to apply established software engineering methodologies from the initial requirements gathering and analysis, through system architecture and detailed design, to implementation, testing, verification, validation, and maintenance. The project should reflect a systematic and disciplined approach to software development. 2. **Consideration of Constraints:** Your solution must demonstrate thoughtful consideration of relevant constraints, which may include health and safety, environmental impact, societal needs, adherence to standards, and other interdisciplinary factors pertinent to your project context. 3. **Comprehensive Documentation:** You are expected to produce and maintain clear, comprehensive, and well-organized documentation for your software system. This includes, but is not limited to, requirements specifications, design documents, test plans, user manuals, and maintenance guides. The project codebase must be maintained in a structured, version-controlled repository with meaningful commit messages, clear directory organization, and thorough inline code comments. All documentation and code should be prepared with future development in mind, enabling subsequent teams to readily understand, maintain, and extend the platform without reliance on external sources or prior team members. Regular updates to both documentation and the repository are expected throughout the project to ensure accuracy and continuity for future contributors. 4. **Demonstration of Artifacts:** Throughout the project, you will be required to present various software artifacts during team meetings. These may include prototypes, design sketches, test harnesses, infrastructure components, and code repositories. Regular demonstrations will be used to assess progress and provide feedback. Successful completion of this capstone project will require not only technical proficiency but also the ability to integrate engineering principles with real-world constraints, communicate effectively through documentation, and engage in collaborative development practices. These expectations align with professional standards in the software engineering discipline. **References:** References related to threat modeling and threat modeling tools will be provided. **Prerequisites:** Enrolment in Software Engineering; proficiency in software design and development; familiarity with the software development lifecycle (SDLC); basic knowledge of system security concepts; experience with version control systems (e.g., Git); teamwork and communication skills; analytical and problem-solving skills.

Program(s)	Software
Maximum number of students	5
Meeting time with supervisor (optional)	Students undertaking this project must be available to meet to discuss project progress, address challenges, and collaborate effectively with team members and the project supervisor. Regular meetings are expected to be primarily in person during regular working hours (Monday-Friday 9:00AM-4:00PM) held every 1-2 weeks for 30-60 minutes.
Do you want the student to contact you before the office assign this project to them ? (Yes/No)	Yes