SYSC 4907 – Fall 2026/Winter 2027

Supervisor	Jason Jaskolka
Co-supervisor	
Course section	SYSC 4907 H
Project ID	H3
Project title	Compass Toolkit: Secure System Analysis Solutions (Multiple Projects)
Project description	Are you interested in building innovative tools that help shape the future of secure software engineering? This capstone project offers you the opportunity to design and develop a cutting-edge software tool that will be integrated into *Compass*, a comprehensive, service-oriented toolkit created by the Cyber Security Evaluation and Assurance (CyberSEA) Research Lab to support secure system design and analysis. Modern software architects, developers, and evaluators face increasing pressure to address security concerns early in the software development lifecycle (SDLC). Yet they often lack integrated, practical tools to measure, evaluate, and improve system security as part of their daily workflow. *Compass* is designed to fill this gap by providing a unified platform for heterogeneous security tools, enabling seamless evaluation and assurance activities throughout the software development lifecycle. As a project team, you'll select a real-world security evaluation, analysis, or design challenge and build a specialized tool to address it. You are encouraged to propose your own ideas or to choose from a wide range of impactful topics, with guidance and approval from the project supervisor. Here are some engaging directions your project could take: • *Security Evaluation*: Build a tool that calculates and visualizes key security metrics for software design artifacts, such as design documents or system models. • *Security Assurance*: Create a solution that provides early assurance of software system security properties using formal analysis tools and techniques. • *Vulnerability Identification*: Develop a tool to detect and classify vulnerabilities in software design artifacts, helping teams address weaknesses before implementation. • *Threat Analysis*: Design a system for analyzing threats to software systems, including features like attack tree analysis, attack simulation, or misuse case development. • *Attacker Behaviour Analysis*: Construct a tool that characterizes attacker behaviours, such as identifying attacker intent or detecting attack patterns. • *Security Design Patterns*: Build a solution to analyze, verify, and validate the effectiveness of security design patterns in mitigating threats and enhancing system security at design time. • *Security Requirements Analysis*: Develop a tool for analyzing and assessing critical security and compliance requirements, including conflict identification, risk management, adequacy measurement, and traceability support.

- *Security Process and Procedure Analysis*: Create a tool to evaluate and improve security processes and procedures, such as incident response plans, security policies, and compliance workflows.
- *System Security Testing*: Develop a solution for generating security test cases from software requirements and design artifacts, designing security test plans, and calculating test coverage to ensure robust system security validation.

Feel free to propose your own innovative ideas; projects that address emerging or unique security challenges are always welcome!

Project teams will have the flexibility to choose any technology stack that supports deploying the developed tool as a cloud-native web application. Since Compass uses Docker for containerization, all tools must be packaged as Docker containers to enable consistent deployment and management within the platform. The container images must be uploaded to a container registry accessible by Compass, which handles deployment, access control, and availability of the tools. Tools that meet the quality standards set by the project supervisor will be featured on the Compass Explore page, alongside other existing tools.

Project Objectives:
After selecting the security evaluation, analysis, or design challenge, the project will aim to:

1. Identify and document a suitable approach for analyzing the chosen software artifacts, providing clear rationale for the selected method.
2. Design and implement robust software mechanisms and tools to automate the chosen analysis process, ensuring accuracy, efficiency, and usability.
3. Create user-friendly reporting features that present analysis results in a way that directly supports system architects, developers, evaluators, and researchers in making informed security decisions.
4. Package the developed tool as a Docker container and successfully deploy it on the Compass platform, ensuring seamless integration, accessibility, and maintainability.

Project Expectations:
This project is open to students enrolled in the Software Engineering program.

1. **Application of Software Engineering Principles:** Students 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 appropriate documentation for your software system. This includes, but is not limited to, requirements specifications, design documents, test plans, user manuals, and maintenance guides. 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 the 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:** Relevant references and resources, including Compass contribution guides and documentation, 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); basic knowledge of cloud-native application development and containerization (e.g., Docker); 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