Final project- 2025/2026 3GL

Integrating Generative AI into DevSecOps for Automated Security Policy Generation

1. Context & problem statement

Modern software development increasingly relies on DevSecOps pipelines to ensure continuous integration of security throughout the development lifecycle.
However, one persistent challenge remains: translating technical vulnerability reports (SAST, SCA, DAST) into actionable, human-readable security policies aligned with international standards such as NIST CSF and ISO/IEC 27001.

The recent emergence of Large Language Models (LLMs) (e.g., LLaMA 3, DeepSeek R1, GPT-type models) offers a new opportunity to automate this translation process and create dynamic, adaptive, and standards-compliant security documentation.

This project explores how generative AI can assist in transforming DevSecOps outputs into structured security policies, bridging the gap between technical detection and organizational governance.

2. Project Objectives

By the end of this project, students should be able to:

1. Understand the role of DevSecOps in modern secure software development.
2. Apply AI-assisted automation to enhance the security lifecycle.
3. Use SAST, SCA, and DAST tools to identify vulnerabilities.
4. Develop parsing mechanisms to preprocess security reports.
5. Employ LLMs for automatic security policy generation conforming to frameworks such as NIST CSF or ISO 27001.

Bonus:

6. Conduct a research-level evaluation using linguistic and structural metrics (e.g., BLEU, ROUGE-L).
7. Critically analyze ethical, privacy, and reliability concerns when using AI in security governance.

3. Required Background

Students should have prior exposure to:

- Cybersecurity fundamentals and risk management.
- DevSecOps concepts (CI/CD, security automation, IaC).
- Basic knowledge of AI and prompt engineering.
- Familiarity with cloud-based pipelines (AWS/GitLab/GitHub).

4. Project Description

This project consists of two main components:

A. Technical Implementation

- Set up a DevSecOps pipeline integrating SAST, SCA, and DAST tools.
- Collect and structure vulnerability reports (in JSON, XML, or HTML).
- Design a rule-based parser to preprocess these reports.
- Use LLMs (e.g., LLaMA 3.3, DeepSeek R1) to generate draft security policies linked to detected vulnerabilities (You should think of a comparative study).

Bonus

- Implement a second-stage refinement model to harmonize and validate the policy content.

B. Research and Evaluation

- Analyze how effectively LLMs interpret security data and generate compliant policies.
- Compare generated policies with reference NIST/ISO templates.
- Use BLEU and ROUGE-L metrics to assess similarity, fluency, and compliance.
- Discuss the implications of using generative AI for automated governance, traceability, and explainability.

5. Expected Tasks

Task	Description
1	Literature review on DevSecOps, policy automation, and LLMs in cybersecurity
2	Setup of CI/CD pipeline and integration of SAST/SCA/DAST
3	Development of rule-based parsing script
4	Prompt engineering and LLM-based policy generation
5	Evaluation of generated policies with BLEU and ROUGE-L
6	Final report with technical results, analysis, and ethical discussion

6. Deliverables

1. Project Report

   - ✓ Introduction & Context
   - ✓ Architecture & Implementation
   - ✓ Results & Evaluation
   - ✓ Discussion & Future Work

2. Demonstration or Prototype

   - ✓ Functional pipeline or simulation showing AI-assisted policy generation.

3. Presentation (10–15 min)

   - ✓ Overview of methodology, findings, and reflections.

7. Evaluation Criteria

Criterion	Weight
Technical Implementation	25%
Research and Analysis	20%
Quality of Generated Policies (metrics)	20%
Report Structure and Clarity	15%
Presentation and Discussion	20%

8. Suggested Tools and Frameworks

- CI/CD & DevSecOps: GitLab CI/CD, GitHub Actions, Jenkins, or AWS CodePipeline.
- Security Tools: SonarQube (SAST), OWASP Dependency-Check (SCA), OWASP ZAP (DAST).
- LLM Integration: Hugging Face models (LLaMA 3, DeepSeek R1, GPT-based APIs).
- Evaluation Metrics: BLEU, ROUGE-L (Python libraries).

10. Outcome

At the end of this project, students will:

- Produce a proof-of-concept pipeline that transforms technical vulnerability outputs into high-level security documentation.
- Contribute to research on AI-driven governance and compliance.
- Demonstrate the ability to bridge engineering practice with scientific inquiry.