# Cyber Security Projects for Final Year Students

List of Proposals

# Table of Contents

## 1. AI-Powered Detections-as-Code in specific SIEM/SOAR solutions

**Description:**

In the field of cybersecurity, we constantly face the challenge of speed: the speed of threat evolution versus the speed of our deployment. This project targets a blueprint for engineering; the next generation SOC focused entirely on automation and governance. We need to move beyond basic CI/CD by fully embracing a GitOps-driven pipeline, ensuring a structured, fully automated process for security rule development. Proposed solution must demonstrate how to eliminate the manual steps that lead to analyst burnout.

The technical part must focus on building a secure detection factory, by implementing GitOps and AI to achieve massive efficiency and governance:

- GITOPS for Detection Logic (KQL and SOAR) with Concept, Version Control & Governance
- GITAIOPS for AI Agent integration to achieve automated validation, code quality and documentation on the fly.

**Complexity:** High

## 2. Next-Gen Secure Development: Testing Code, Dependencies & AI Models Together

**Description:**

Unify SAST, SCA, DAST and AI checks in a single DevSecOps pipeline that validates code, open-source libraries and AI components together.

Objective here is to correlate results from these diverse testing tools to identify overlapping vulnerabilities, prioritize risks, and reduce false positives. The project involves designing and implementing automated workflows that run security scans at appropriate stages of the CI/CD process, aggregating findings into a unified dashboard or report for clear, actionable insights.

Specifically, students are expected to:

- Integrate SAST, SCA, DAST, and AI model validation tools within a DevSecOps pipeline environment.

- Develop methods to correlate and consolidate results from multiple scanners and AI checks. Policy based, rule based solution for vulnerabilities correlation.
- Implement reporting mechanisms that highlight critical vulnerabilities, affected components, and remediation guidance. Assess AI agent potential for reporting needs.
- Test the pipeline with real-world codebases and AI models, demonstrating improved visibility and risk management over using isolated tools.
- Ensure scalability and maintainability of the pipeline for ongoing software development cycles.
- This project will equip students with practical skills in modern secure development practices, bridging traditional software security with emerging AI model assurance needs.

**Complexity:** High – could relate to no. 3

## 3. MCPs as Security Co-Pilots: Automated AI Audits

**Description:**

How to use Model Context Protocol (MCP) to automatically inspect, audit and validate AI systems as part of your DevSecOps workflows.

Students will develop an automated auditing framework that integrates MCP into existing DevSecOps workflows, enabling continuous verification of AI model integrity, compliance, and performance throughout the development lifecycle.

Specifically, students are expected to:

- Research and understand the Model Context Protocol (MCP) specifications and its role in AI system governance.
- Design and implement tools or scripts that extract MCP metadata from AI models and their development environments.
- Integrate MCP-based audits into CI/CD pipelines to automatically validate AI models against security, fairness, and compliance criteria.
- Develop reporting and alerting mechanisms that provide actionable insights to developers and security teams.
- Develop an AI Bill of Materials (AI-BOM similar to classical SBOM) report.
- Demonstrate the framework using example AI models, showcasing how MCP audits can identify anomalies, data drift, or policy violations early in the development process.

**Complexity:** High – think about dividing it in several subprojects

## 4. AI powered Cyber Threat Intelligence – practical approach

**Description:**

With the steadily growing number of cyberattacks by organized cybercriminals and state actors, it is becoming increasingly difficult to maintain an overview of the threat situation.

The sheer volume of daily threat reports, new vulnerabilities, and attack vectors poses major challenges for security managers. Furthermore, competent specialists capable of processing all this information and deriving appropriate measures from it remain a rare commodity.

The rapid development in the field of AI, and especially in large language models (LLMs), offers fascinating new opportunities in a wide variety of disciplines, also in the context of cybersecurity. In this research project we suggest focusing on the topic of threat intelligence in conjunction with AI.

Using several use cases to demonstrate how customized chatbots can be developed as threat intelligence assistants that process complex data sources such as reports or leak data quickly and precisely. Provide a practical explanation of how sound data preparation and the use of powerful frameworks create the foundation for powerful AI assistants.

**Complexity:** High

## 5. Autonomous Pentesting in CI/CD Pipeline: Continuous Attacks for Continuous Security

**Description:**

Integrate automated recon, scanning and safe exploitation into GitLab CI/CD

Students are expected to:

- Research and select appropriate open-source or commercial tools for automated recon, scanning, and safe exploitation that can be integrated into CI/CD workflows.
- Configure and embed these tools into (GitLab) pipelines, ensuring seamless execution at defined stages such as build, test, or pre-deployment.
- Develop mechanisms to safely simulate exploitation without causing harm to the environment or production data.
- Design automated reporting that consolidates findings, prioritizes vulnerabilities, and provides actionable remediation guidance to developers and security teams.
- Validate the pipeline by testing it on sample applications, demonstrating how continuous pentesting improves security visibility and reduces risk over time.
- Address pipeline performance and security concerns to ensure the automated tests do not disrupt the development process.

**Complexity:** Medium/High

## 6. IoT Deception Honeypot Network

**Description:**

In this project, students should build a virtual honeypot network made up of simulated, low-interaction IoT devices such as smart bulbs, sensors, or cameras. Using tools like Honeyd, Cowrie, or custom Python scripts, they emulate device behavior and monitor how potential attackers interact with the system. All activity—IP addresses, payloads, and attempted exploits—is logged and visualized on a dashboard to reveal attack patterns. Students also examine any captured malware or exploit techniques.

The setup can mimic realistic scenarios, including smart-home intrusions or industrial espionage attempts. By leveraging deception as a defensive strategy, students demonstrate their understanding of proactive cybersecurity and threat intelligence. The project can be further enhanced with attack geolocation and automated reporting to create a comprehensive threat-analysis platform.

**Complexity:** High

## 7. Ransomware Detection and Containment Engine

**Description:**

Develop a ransomware-focused honeynet that identifies suspicious file-encryption activity through file-integrity monitoring. The system places decoy files throughout the environment and alerts on any unauthorized changes or cryptographic operations. When malicious behavior is detected, it can activate containment measures such as isolating affected machines or disabling shared network drives. All attack attempts are logged, visualized on dashboards, and can be replayed for further investigation. This provides a practical platform for organizations or research labs to evaluate, strengthen, and validate their ransomware-mitigation strategies.

**Complexity:** Medium

## 8. Automated Incident Response System

**Description:**

Create an automated, playbook-driven incident response system that reacts to predefined security events such as malware detections or brute-force attempts. The tool integrates with IDS/IPS logs, SIEM platforms, and external threat intelligence sources. Response logic is defined using Python and YAML playbooks, enabling automated actions like isolating compromised endpoints, blocking malicious IPs, or alerting administrators. A management interface allows users to build, test, and refine response workflows. This project is highly valuable for SOC teams, helping streamline operations and significantly reduce incident response times.

**Complexity:** High

## 9. Packet Sniffer for specific IoT protocols (OPC UA protocol, PROFINET, MQTT) to analyse and detect side channel attacks

**Description:**

This project involves developing a packet-analysis tool tailored to IoT and industrial communication protocols such as OPC UA, PROFINET, and MQTT. The system captures and inspects network traffic to help identify unusual patterns, misconfigurations, or potential security weaknesses within these environments. Built-in protocol awareness allows the tool to decode relevant message structures and provide meaningful insights for monitoring or research.

In addition, the project explores side-channel analysis from a defensive and diagnostic perspective—examining how non-network indicators (e.g., timing patterns or resource-usage anomalies) might unintentionally reveal information about device behavior. The focus is on understanding these risks so that better protections and mitigation strategies can be designed.

**Complexity:** Medium/High

## 10. Hardware Gateway for Securing IoT and Industrial Cameras Against Network Intrusions (monitoring the traffic in OT/IoT to detect indicators of compromise (IoCs) and integrate them in SOC/SIEM)

**Description:**

Students will design a hardware-based gateway that captures, monitors and filters network traffic to protect IoT and industrial camera systems from intrusion attempts. The project involves analyzing common vulnerabilities in connected video devices, building a prototype security appliance, and evaluating its effectiveness within a controlled operational technology (OT) or automation environment. The gateway will detect indicators of compromise (IoCs) and support integration with SOC/SIEM platforms for centralized reporting and threat visibility.

**Complexity:** High

# 11. Active Defense Mechanisms for Industrial Robots: Design of a Physical Intrusion-Prevention Device for Automated Production Lines

**Description:**

This thesis examines cybersecurity vulnerabilities in industrial robots used on automated production lines and analyzes how disruptions or unauthorized command flows could affect manufacturing processes. Building on this assessment, students will design a platform-based active-defense system that protects robotic operations without requiring software updates on legacy machines or changes to existing production setups.

The solution uses a centralized cloud environment as a trusted synchronization and management layer. All configuration updates, files, and process data are first validated, scanned, and logged in this shared cloud location before they are allowed to reach the robots. This architecture enables secure file synchronization, protocol translation (e.g., SMBv1 → SMBv3), virus scanning and rule-based file validation, and directory synchronization while providing full visibility into update status across OT networks.

The system maintains strict decoupling between IT and OT networks, supports secure remote access, and allows safe process-data transfer from both modern and legacy equipment. It also integrates seamlessly with SOC/SIEM platforms for centralized monitoring, incident reporting, and detection of indicators of compromise (IoCs). Through the combination of cloud security, industrial automation, robotics, and network-segmentation principles, the project delivers a practical active-defense concept that enhances the cybersecurity resilience of smart manufacturing environments without imposing downtime or costly modernization.

**Complexity:** High – think about dividing it in several subprojects