Capstone Project

To complete this template, please refer to the "Capstone Project Guidelines." Fill in only the necessary sections according to the considerations outlined in the Capstone document.

Creator's name: Arturo Garcia Martin del Campo
Discipline: Security **Level:** General
Problem definition: In today's world, the development of secure software applications is paramount to protect sensitive information, maintain user trust, and ensure the resilience of digital systems against malicious actors. Despite the growing awareness of cybersecurity threats, many security practices are considered in a late phase, increasing the required efforts for solving it and reducing the amount of risks and vulnerabilities that can be managed effectively. This gap between the evolving threat landscape and secure development practices highlights the critical need for professionals who possess both practical skills and a deep understanding of secure software development from the beginning.
Practical Application: This capstone project aims to address this challenge by providing a comprehensive, hands-on experience in secure software development. The project is divided into three distinct sections, each focusing on a vital aspect of secure software development: • Introduction to Application Security and Best Practices in Coding   ○ In this section, students will create a secure development environment by configuring and employing various security tools, including vulnerability scanning tools.

- The primary objective is to understand the findings generated by these tools and how they contribute to the development of secure software.
- Students will gain insights into the importance of secure coding practices, tool integration, and the role of automation in identifying and mitigating security vulnerabilities early in the development lifecycle.

- Section 2: Secure Coding Techniques
  - Building upon the foundation established in the first section, students will embark on the process of coding a secure application.
  - They will apply secure coding principles, including input validation, access control, authentication, and encryption, to develop an application that is resilient to common security threats.
  - By creating a functional secure application, students will gain practical experience in applying security concepts in real-world scenarios.
- Risk Assessment and Vulnerability Management
  - The final section of the capstone project involves performing a comprehensive risk assessment of the secure application developed in Section 2.
  - Students will identify potential security risks, prioritize them based on severity and likelihood, and propose mitigation strategies.
  - This section serves as the bridge between theory and practice, allowing students to merge their findings from Sections 1 and 2 into a holistic understanding of secure software development.

Objectives:

- Equip students with a hands-on, practical understanding of secure development practices.
- Develop the ability to leverage security tools effectively for vulnerability identification and mitigation.
- Cultivate the skills needed to code secure applications resilient to common threats.
- Foster a comprehensive understanding of risk assessment and its role in secure software development.

By the end of this capstone project, participants will be well-prepared to enter the workforce as secure software developers, armed with the knowledge, skills, and experience needed to create robust and secure applications in today's digital landscape.

General Description:

This capstone project focuses on the comprehensive exploration of secure software development. Divided into three sections, it encompasses the setup of a secure development environment, the creation of a secure application, and the performance of a risk assessment. Students will gain practical skills and knowledge to develop secure software while understanding the importance of tools, coding practices, and risk management in the process.

Deliverable description	**Milestone Integration** (module, sections it aligns with)	**Scoring integration**	**Submission Format**	**Deadline**
1 GitHub Repository - GH	Introduction to Application Security and Best Practices in Coding Secure Coding Techniques Risk Assessment and Vulnerability Management	50	GitHub	Oct 20, 2023
2 Initial Vulnerability Report - IVR	Introduction to Application Security and Best Practices in Coding	10	PDF	Nov 3, 2023
3 Final Vulnerability Report - FVR	Risk Assessment and Vulnerability Management	40	PDF	Nov 17, 2023

Módulo	**Requerimiento**	**Entregable**	**Score**
Introduction to Application Security and Best Practices in Coding	Escanea el código https://github.com/TsuyoshiUshio/VulnerableApp con SonarQube	IVR	5
	Analizar a detalle los hallazgos de SonarQube, utiliza las secciones	IVR	5

	What is the risk, *Are you at risk?* y *How can you fix it* para desarrollar un reporte con la siguiente información: • Descripción de las 3 vulnerabilidades que consideres más importantes. • Líneas de código afectadas por las vulnerabilidades afectadas • Propuesta de remediación • Justificación de 3 falsos positivos (Hallazgos que en realidad son falsos positivos) Usa el siguiente template Vulnerability Management R…		
	Instalar y configurar correctamente Sonarlint en un entorno de Visual Studio	GH	3
	Automatiza el análisis en el repositorio que utilizarás para el proyecto final con SonarQube utilizando GitHub Actions	GH	4
	Habilita Dependabot en tu repositorio de GitHub	GH	3
Secure Coding Techniques	Desarrolla una aplicación con el	GH	6

	lenguaje de tu preferencia, donde le des la oportunidad al usuario de introducir cadenas de texto.		
	Desarrolla una función para sanitizar el texto introducido por el usuario, permitiendo solo caracteres no maliciosos. Es sugerible utilizar expresiones regulares.	GH	7
	Detecta números de tarjetas de crédito/débito en la cadena isanitizada y enmascáralos, permitiendo ver solo los últimos 4 números.	GH	6
	Desarrolla un a función para calcular el hash en SHA256 de la cadena sanitizada y enmascarada.	GH	7
	Desarrolla una función para cifrar en AES256 la cadena sanitizada y enmascarada.	GH	7
	Desarrolla una función para descifrar la cadena cifrada, utiliza la función para calcular el hash y confirma que la cadena descifrada siga resultando en el mismo hash.	GH	7
Risk Assessment and Vulnerability Management	Utiliza Microsoft Threat Modelling Tool para detectar riesgos en la	FVR	8

	arquitectura compartida por Wizeline.		
	Utiliza el navegador MITRE ATT&CK para detectar riesgos de acuerdo a los grupos cibercriminales que pudieran ser de interés para el contexto de tu desarrollo	FVR	8
	Haz un reporte de acuerdo a los hallazgos que hayas tenido con SonarQube, dependabot y Microsoft Threat Modelling Tool: • Descarta y justifica los falsos positivos. • Documenta el detalle de las vulnerabilidades y riesgos detectados. • Identifica si alguno de los hallazgos de estas herramientas, está relacionado a alguno de los hallazgos que tuviste con MITRE ATT&CK. • Haz un plan de remediación, priorizando su riesgo de acuerdo a MITRE ATT&CK y la criticidad que las otras herramientas dieron Usa el siguiente template Vulnerability Management R…	FVR	24

Documentation needed

- https://docs.sonarsource.com/sonarqube/latest/setup-and-upgrade/install-the-server/
- https://docs.sonarsource.com/sonarlint/intellij/getting-started/installation/
- https://learn.microsoft.com/es-es/azure/security/develop/threat-modeling-tool-feature-overview
- How to use the MITRE ATT&CK Navigator

Tools/Technologies

- Java JRE
- SonarQube
- SonarLint
- Microsoft Threat Modeling Tool
- GitHub
- Regexr
- MITRE ATT&CK Navigator
- Visual Studio
- Any Database

Presentation Components*

Collaboration guidelines

Quality standards

Attribution and citation

