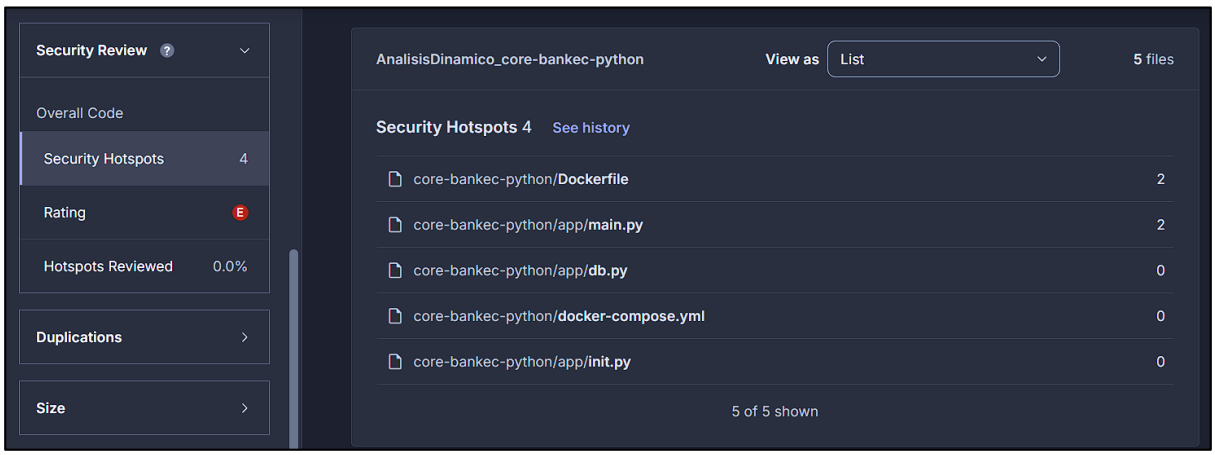
INFORME DE ANÁLISIS ESTÁTICO
Grupo 4

Elaborado por:

- Miguel Guilca
- Mateo Dueñas
- Pamela Cruz
- Christopher Zambrano

Resumen ejecutivo:

Para este caso se utilizó la herramienta SonarQube Cloud, la cual fue conectada a un repositorio con el código fuente del aplicativo, la misma encontró 4 puntos críticos de seguridad visibles en el siguiente gráfico.



La puntuación de severidad fue obtenida utilizando la calculadora “Common Vulnerability Scoring System Version 4.0 Calculator” ubicada en la página web de FIRST en el siguiente enlace: https://www.first.org/cvss/calculator/4-0.

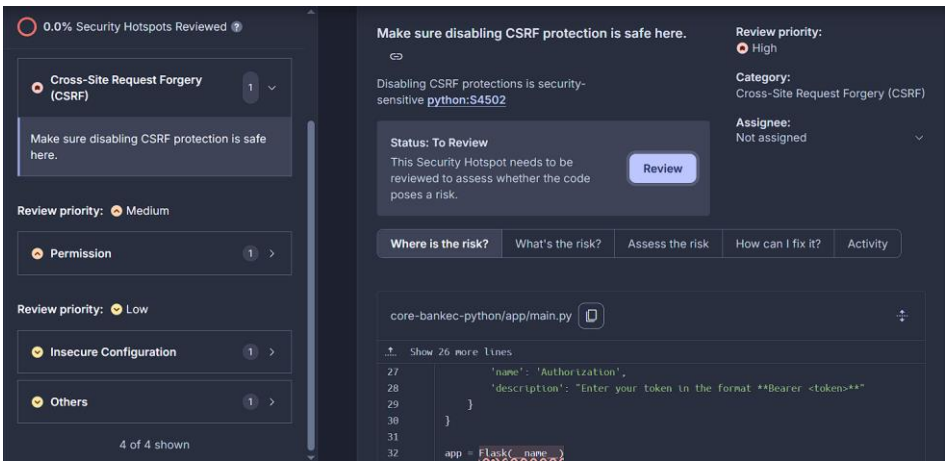
Considerando las rubricas de puntuación que la misma página web facilita, visibles en el siguiente enlace: https://www.first.org/cvss/v4-0/user-guide#Scoring-Rubrics

Hallazgos:

Identificador	PCRIT-001: Desactivación del CSRF
Descripción	Un ataque de falsificación de solicitud entre sitios (CSRF) ocurre cuando un atacante puede obligar a un usuario confiable de una aplicación web a realizar acciones sensibles que no pretendía, como actualizar su perfil o enviar un mensaje, o más generalmente cualquier cosa que pueda cambiar el estado de la aplicación.
Severidad	Common Vulnerability Scoring System Version 4.0 Calculator CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:A/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N Reset CVSS v4.0 Score: 7.4 / High
Referencias	• CWE - CWE-352 - Cross-Site Request Forgery (CSRF) • OWASP - Cross-Site Request Forgery

Recomendación	Ya que esta es una aplicación ***Flask***, el módulo "*CSRFProtect"* debe ser utilizado y no deshabilitado con "WTF_CSRF_ENABLED" configurado en *false,* además de evitar deshabilitarlo en vistas o forms específicos.
	`app = Flask(__name__)`
	`csrf = CSRFProtect()`
	`csrf.init_app(app)`
	
	`@app.route('/example/', methods=['POST']) # Compliant`
	`def example():`
	`  return 'example '`
	
	`class unprotectedForm(FlaskForm):`
	`  class Meta:`
	`  csrf = True`
	
	`  name = TextField('name')`
	`  submit = SubmitField('submit')`

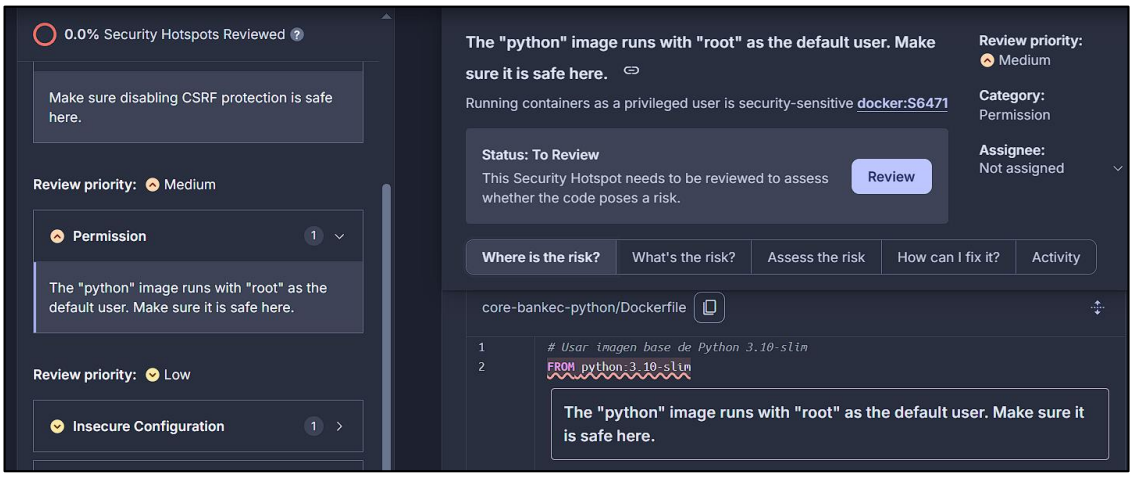
Evidencia:



Identificador	PCRIT-002: Contenedor ejecutándose con usuario root por defecto
Descripción	Ejecutar contenedores como un usuario con privilegios debilita su seguridad en tiempo de ejecución, lo que permite que cualquier usuario cuyo código se ejecute en el contenedor realice acciones administrativas. En los contenedores de Linux, el usuario con privilegios suele llamarse *root*. En los contenedores de Windows, el equivalente es *ContainerAdministrator*.
Severidad	Common Vulnerability Scoring System Version 4.0 Calculator CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:P/VC:H/VI:H/VA:N/SC:L/SI:L/SA:L Reset CVSS v4.0 Score: 8.5 / High
Referencias	• CWE - CWE-284 - Improper Access Control • nginxinc/nginx-unprivileged: Example of a non-root container by default

Recomendación	Crear un nuevo usuario predeterminado y úsarlo con la instrucción USER. Algunos administradores de contenedores crean un usuario específico sin configurarlo explícitamente como predeterminado, como postgresql o zookeeper. Se recomienda usar estos usuarios en lugar de root.
	`FROM alpine`
	`RUN addgroup -S nonroot \`
	`&& adduser -S nonroot -G nonroot`
	`USER nonroot`
	`ENTRYPOINT ["id"]`

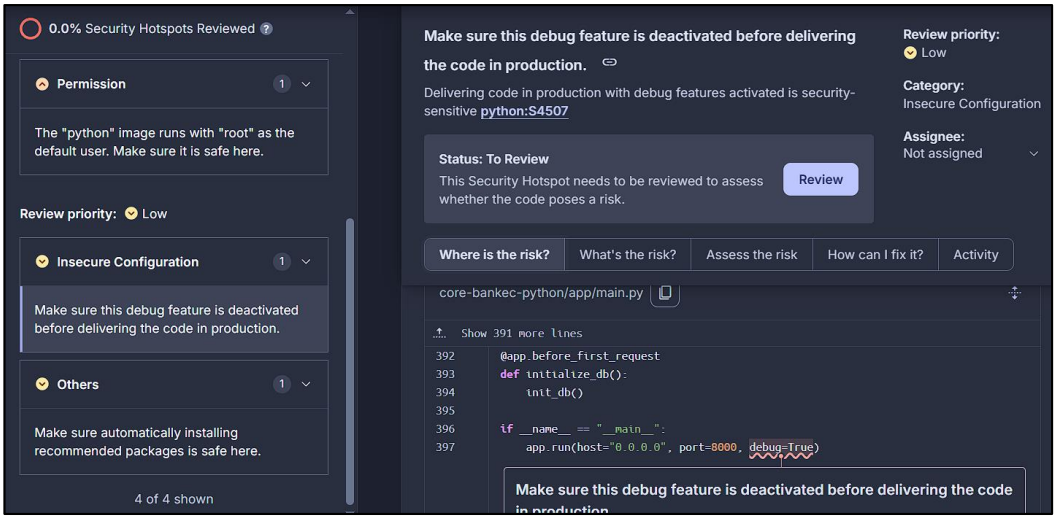
Evidencia:



Identificador	PCRIT-003: Modo de depuración activado
Descripción	Las herramientas y los frameworks de desarrollo suelen incluir opciones para facilitar la depuración a los desarrolladores. Si bien estas funciones son útiles durante el desarrollo, nunca deben habilitarse para aplicaciones implementadas en producción. Las instrucciones de depuración o los mensajes de error pueden filtrar información detallada sobre el sistema, como la ruta de la aplicación o los nombres de archivo.
Severidad	Common Vulnerability Scoring System Version 4.0 Calculator CVSS:4.0/AV:N/AC:L/AT:P/PR:L/UI:N/VC:L/VI:L/VA:L/SC:L/SI:L/SA:L Reset CVSS v4.0 Score: 2.3 / Low
Referencias	• CWE - CWE-489 - Active Debug Code • CWE - CWE-215 - Information Exposure Through Debug Information

Recomendación	No activar las funciones de depuración en servidores de producción o aplicaciones distribuidas a usuarios finales.
	`from flask import Flask`
	`app = Flask()`
	`app.debug = False`
	`app.run(debug=False)`

Evidencia:



Identificador	PCRIT-004: Instalación automática de paquetes no estrictamente requeridos
Descripción	Instalar automáticamente los paquetes recomendados puede generar vulnerabilidades en la imagen de Docker, ya que, los paquetes potencialmente innecesarios se instalan mediante un gestor de paquetes Debian conocido. Estos paquetes aumentan la superficie de ataque del contenedor creado, ya que podrían contener vulnerabilidades no identificadas o código malicioso. En general, cuantos más paquetes se instalan en un contenedor, más débil es su seguridad. Dependiendo de las vulnerabilidades introducidas, un atacante que acceda a dicho contenedor podría utilizarlas para escalar privilegios. Eliminar los paquetes no utilizados también puede reducir significativamente el tamaño de la imagen de Docker.
Severidad	Common Vulnerability Scoring System Version 4.0 Calculator CVSS:4.0/AV:N/AC:H/AT:P/PR:H/UI:N/VC:N/VI:N/VA:N/SC:L/SI:L/SA:L Reset CVSS v4.0 Score: 2.1 / Low
Referencias	• Debian Documentation - Binary Dependencies • Ubuntu Blog - Container size reduction

Recomendación	Evitar la instalación de paquetes de dependencias no estrictamente requeridos.
	`FROM ubuntu:22_04`
	`RUN apt --no-install-recommends install -y build-essential`
	`RUN apt-get --no-install-recommends install -y build-essential`
	`RUN aptitude --without-recommends install -y build-essential`

Evidencia: