

ESCUELA POLITÉCNICA NACIONAL



**CARRERA DE INGENIERÍA DE
SOFTWARE**

**DESARROLLO DE SOFTWARE
SEGURO ISWD853**

Informe de análisis estático

Alumnos:

Alejandro Jiménez

Bryan Rosillo

Christian Hernández

Jorge Segovia

Mateo Dávalos

PROFESOR: Jhonattan Barriga

FECHA DE ENTREGA: 08/01/2025

Contenido

I. Código fuente3

II. Resumen ejecutivo3

III. Informe y análisis.....5

 a) Resultados del análisis estático5

I. Código fuente

- https://github.com/BryanRosillo/Core_bank_ec

II. Resumen ejecutivo

En este apartado se resumirá los hallazgos que se han dado en el análisis de la aplicación analizada, en la siguiente imagen se puede observar de forma gráfica las vulnerabilidades halladas.

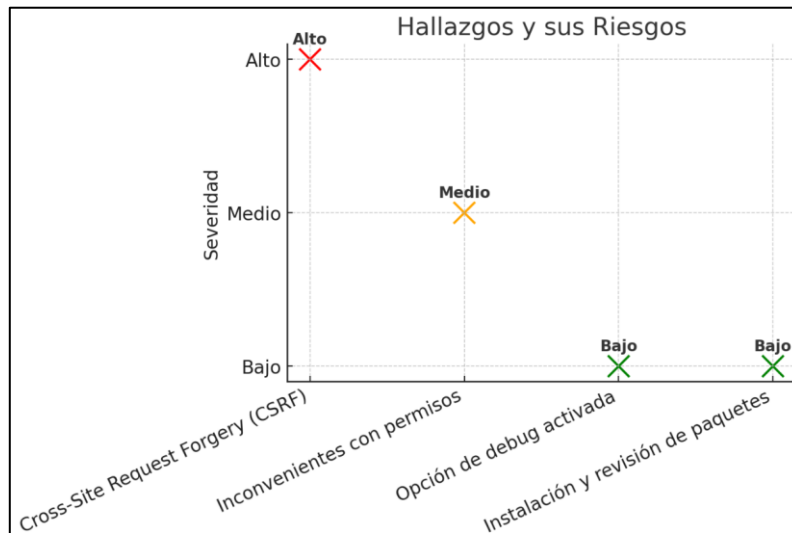


Figura 1. Gráfica de resumen de vulnerabilidades

Ahora se presenta a manera de tablas la gravedad y recomendaciones para combatir con dichas vulnerabilidades.

0	1	1	2	0
Crítico	Alto	Medio	Bajo	Informativo

Tabla 1. Vulnerabilidades descubiertas y su gravedad

Hallazgo	Severidad	Recomendación
Cross-Site Request Forgery (CSRF)	Alto	Para proteger contra ataques CSRF, se debe habilitar la protección por defecto en métodos HTTP inseguros, implementar tokens CSRF seguros y evitar el uso de métodos como GET para realizar operaciones sensibles. En aplicaciones Flask, es fundamental utilizar el módulo CSRFProtect y asegurarse de que no se deshabilite, evitando configurar WTF_CSRF_ENABLED = False. Además, la protección CSRF debe mantenerse activa en vistas y formularios (csrf = True). Estas prácticas refuerzan la seguridad de la aplicación y minimizan el riesgo de explotación de vulnerabilidades.

Inconvenientes con permisos.	Medio	En el Dockerfile, se recomienda crear un usuario no privilegiado con la instrucción USER y, si es posible, utilizar usuarios específicos de la imagen, como postgresql o zookeeper, en lugar de root. En contenedores Windows, se debe emplear ContainerUser. Al ejecutar el contenedor, es importante especificar un usuario con la opción --user en Docker o en docker-compose.yml, y en Linux, asignar capacidades específicas solo cuando sea necesario para reducir riesgos de seguridad y minimizar la exposición a ataques.
Opción de debug activada	Bajo	En aplicaciones Flask, es fundamental deshabilitar las funciones de depuración en entornos de producción para evitar la exposición de información sensible. Para ello, se debe configurar app.debug = False y asegurarse de ejecutar la aplicación con app.run(debug=False). Mantener la depuración activada en producción puede revelar rutas de archivos, configuraciones internas y mensajes de error detallados, lo que aumenta el riesgo de explotación de vulnerabilidades. Implementar esta práctica refuerza la seguridad y evita fugas de información crítica
Instalación y revisión de paquetes	Bajo	Para mejorar la seguridad en Docker, es fundamental evitar la instalación de paquetes innecesarios y verificar vulnerabilidades en los paquetes requeridos. Se recomienda utilizar: "RUN apt update && apt --no-install-recommends install -y build-essential && rm -rf /var/lib/apt/lists/*"

Tabla 2. Recomendaciones para las vulnerabilidades

III. Informe y análisis

a) Resultados del análisis estático

Para llevar a cabo el análisis estático, se ha elegido la herramienta SonarCloud, ya que permite un monitoreo continuo ante nuevos cambios o actualizaciones. Además, facilita la identificación de vulnerabilidades de seguridad, proporcionando una trazabilidad clara del origen del problema y las posibles soluciones.

- **Problema 1**

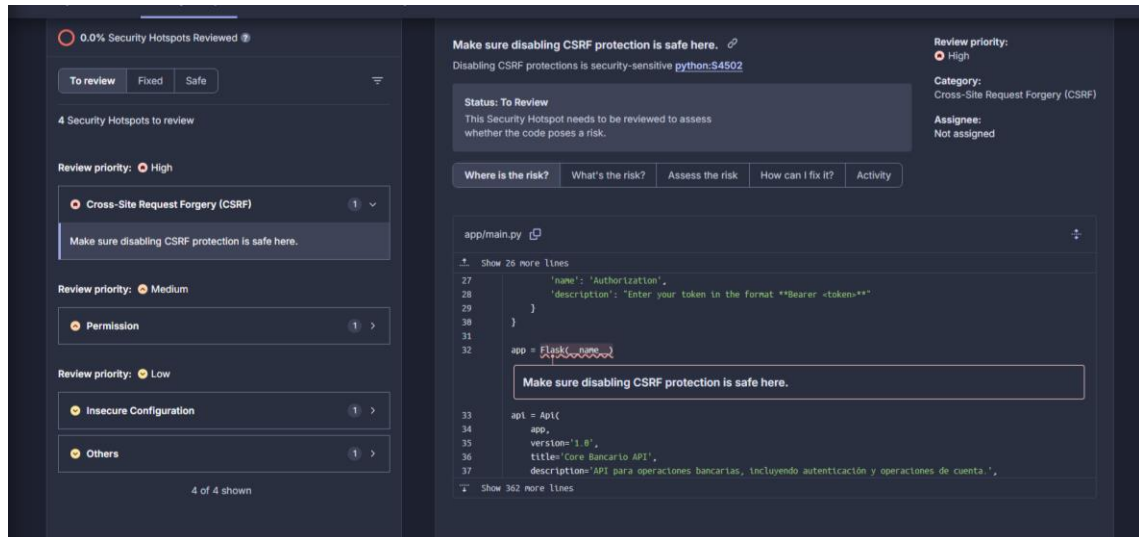


Figura 2. Resultado de problema de seguridad nivel alto

Identificador	Problema de seguridad WT-1
Título	Cross-Site Request Forgery (CSRF)
Descripción	<p>Un ataque de Cross-Site Request Forgery (CSRF) ocurre cuando un atacante logra que un usuario legítimo de una aplicación web ejecute acciones sensibles sin su consentimiento, como actualizar su perfil, enviar un mensaje o cualquier otra operación que modifique el estado de la aplicación.</p> <p>El atacante puede engañar a la víctima para que haga clic en un enlace que desencadena la acción privilegiada o visite un sitio web malicioso que contiene una solicitud web oculta. Dado que los navegadores incluyen automáticamente las cookies en las solicitudes, estas acciones pueden ser autenticadas y ejecutadas sin que el usuario lo advierta, lo que compromete la seguridad de la aplicación.</p>
Referencias	<ul style="list-style-type: none">• OWASP_2021_A07• CWE-352• CWE-20

Riesgo	OWASP Risk Rating Calculator			
	Likelihood Factors		Impact Factors	
	Threat Agent Factors	Vulnerability Factors	Technical Impact Factors	Business Impact Factors
	Skill Level 5 - Advanced computer user	Ease of Discovery 7 - Easy	Loss of Confidentiality 6 - Minimal critical data or extensive non-	Financial Damage 7 - Significant effect on annual profit
	Motive 7	Ease of Exploit 9 - Automated tools available	Loss of Integrity 5 - Extensive digitally corrupt data	Reputation Damage 9 - Brand damage
	Opportunity 4 - Special access or resources required	Awareness 6 - Obvious	Loss of Availability 7 - Extensive primary services interrupted	Non-compliance 5 - Clear violation
	Size 4 - Intranet users	Intrusion Detection 4	Loss of Accountability 7 - Possibly traceable	Privacy Violation 5 - Hundreds of people
	Threat Agent Factor: Medium (TAF: 5)	Vulnerability Factor: High (VF: 6.5)	Technical Impact Factor: High (TIF: 6.25)	Business Impact Factor: High (BIF: 6.5)
	Likelihood Factor: Medium (LF: 5.75)		Impact Factor: High (IF: 6.5)	
	Overall Risk Severity: High			
	Score Vector: (SL:5/M:7/OA:4/SA:4/ED:7/EE:9/A:6/ID:4/LC:6/LI:5/LAV:7/LAC:7/FD:7/RD:9/NC:5/PV:5)			

Recomendaciones:

Por la parte de prácticas se debe tener en cuenta los siguientes puntos:

- Habilitar la protección por defecto en todos los métodos HTTP inseguros.
- Implementar un token CSRF seguro y difícil de predecir para validar cada solicitud que modifique el estado de la aplicación.
- Evitar el uso de métodos HTTP seguros (como GET) para realizar operaciones sensibles, ya que estos deben emplearse exclusivamente para la recuperación de información.

En cuanto al código como se está usando flask se recomienda lo siguiente:

- Se debe utilizar el módulo **CSRFProtect** y asegurarse de que no se deshabilite, evitando configurar **WTF_CSRF_ENABLED** en **False**. Esto garantiza una protección adecuada contra ataques **CSRF** y refuerza la seguridad de las solicitudes en la aplicación.
- No se debe deshabilitar la protección CSRF en las vistas y formularios: **csrf = True # Compliant**

- Problema 2

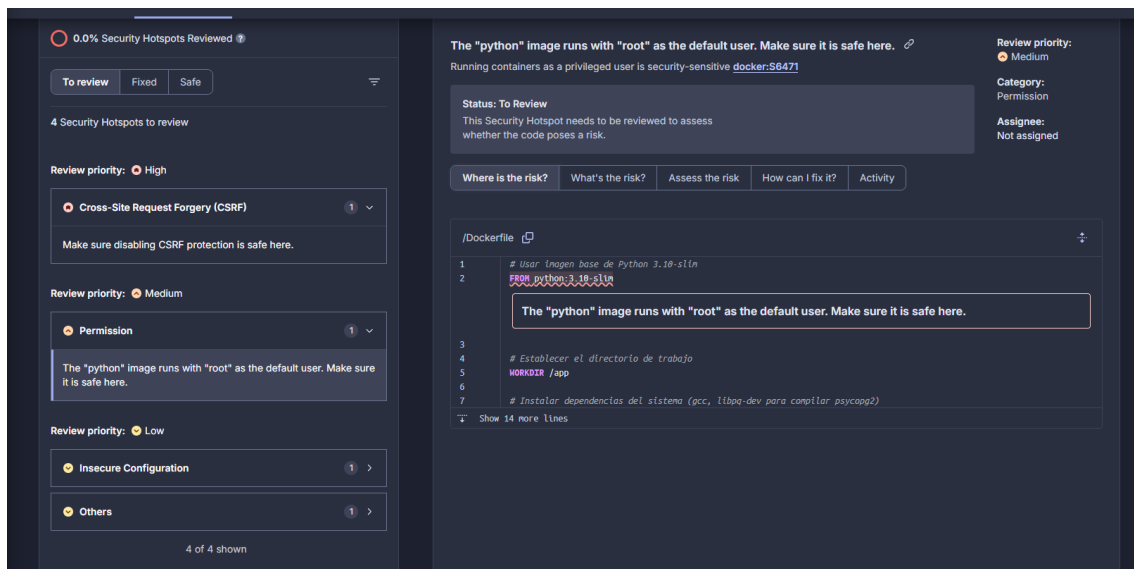


Figura 3. Resultado de problema de seguridad nivel medio

Identificador	Problema de seguridad WT-2
Título	Inconvenientes con permisos.
Descripción	<p>El contenedor de Python se ejecuta con "root" como usuario predeterminado, lo que puede representar un riesgo de seguridad. Ejecutar contenedores con un usuario privilegiado (root en Linux o ContainerAdministrator en Windows) compromete la seguridad, permitiendo que cualquier código en el contenedor realice acciones administrativas.</p> <p>Un atacante puede ejecutar código arbitrario, acceder a archivos sensibles, establecer conexiones maliciosas o incluso escapar del contenedor.</p>
Referencias	<ul style="list-style-type: none"> • OWASP_2021_A05 • CWE-284
Riesgo	

OWASP Risk Rating Calculator

Likelihood Factors

Threat Agent Factors

Skill Level
4
Motive
4 - Possible reward
Opportunity
4 - Special access or resources required
Size
3

Threat Agent Factor:
Medium (TAF: 3.75)

Vulnerability Factors

Ease of Discovery
3 - Difficult
Ease of Exploit
5 - Easy
Awareness
4 - Hidden
Intrusion Detection
4

Vulnerability Factor:
Medium (VF: 4)

Impact Factors

Technical Impact Factors

Loss of Confidentiality
7 - Extensive critical data disclosed
Loss of Integrity
4
Loss of Availability
7 - Extensive primary services interrupted
Loss of Accountability
5

Technical Impact Factor:
Medium (TIF: 5.75)

Business Impact Factors

Financial Damage
7 - Significant effect on annual profit
Reputation Damage
3
Non-compliance
3
Privacy Violation
5 - Hundreds of people

Business Impact Factor:
Medium (BIF: 4.5)

Likelihood Factor: Medium (LF: 3.875)

Impact Factor: Medium (IF: 4.5)

Overall Risk Severity: Medium

Score Vector: (SL:4/MA:0/A:5/S:3/ED:3/EE:5/A:4/ID:4/LC:7/LI:4/LAV:7/LAC:5/FD:7/RD:3/NC:3/PV:5)

Recomendaciones:

Para el dockerfile lo que se puede hacer es:

- Crear un usuario no privilegiado y establecerlo con la instrucción USER.
- Usar usuarios específicos proporcionados por la imagen, como postgresql o zookeeper, en lugar de root.
- En contenedores Windows, utilizar ContainerUser.

Al momento de ejecutar el contenedor pues:

- Especificar un usuario con la opción --user en Docker o en docker-compose.yml.
- Asignar capacidades específicas en Linux solo cuando se requieran privilegios elevados.

Para explicar mejor lo dicho se colocan imágenes con código de ejemplo de algunas de las situaciones:

Linux

```
FROM alpine

RUN addgroup -S nonroot \
  && adduser -S nonroot -G nonroot

USER nonroot

ENTRYPOINT ["id"]
```

Windows

```
FROM mcr.microsoft.com/windows/servercore:ltsc2019

RUN net user /add nonroot

USER nonroot
```


Builds

```
FROM alpine as builder
COPY Makefile ./src /
RUN make build

FROM alpine as runtime
RUN addgroup -S nonroot \
    && adduser -S nonroot -G nonroot
COPY --from=builder bin/production /app
USER nonroot
ENTRYPOINT ["/app/production"]
```

• Problema 3

The screenshot displays a security tool interface with a sidebar on the left and a main content area on the right. The sidebar shows a summary of security hotspots, with 4 hotspots to review. The main content area shows a specific security hotspot titled 'Make sure this debug feature is deactivated before delivering the code in production.' The status is 'To Review'. The category is 'Insecure Configuration'. The review priority is 'Low'. The assignee is 'Not assigned'. The code snippet shows a Python file 'app/main.py' with a 'debug=True' flag in the 'app.run()' function call. A warning message is displayed below the code snippet: 'Make sure this debug feature is deactivated before delivering the code in production.'

Figura 4. Resultado de problema de seguridad nivel bajo

Identificador	Problema de seguridad WT-3
Título	Opción de debug activada
Descripción	Las herramientas y frameworks de desarrollo suelen incluir opciones para facilitar la depuración. Aunque son útiles en entornos de desarrollo, nunca deben estar habilitadas en producción, ya que pueden exponer información sensible del sistema, como rutas de la aplicación o nombres de archivos, a través de mensajes de error o instrucciones de depuración.
Referencias	<ul style="list-style-type: none"> OWASP_2021_A5 CWE-489 CWE-215
Riesgo	

OWASP Risk Rating Calculator

Likelihood Factors

Threat Agent Factors

Skill Level

3 - Network and programming skills

Motive

2

Opportunity

2

Size

1

Threat Agent Factor: Low (TAF: 2)

Likelihood Factor: Low (LF: 2.875)

Vulnerability Factors

Ease of Discovery

3 - Difficult

Ease of Exploit

3 - Difficult

Awareness

5

Intrusion Detection

4

Vulnerability Factor: Medium (VF: 3.75)

Impact Factors

Technical Impact Factors

Loss of Confidentiality

1

Loss of Integrity

2

Loss of Availability

2

Loss of Accountability

3

Technical Impact Factor: Low (TIF: 2)

Business Impact Factors

Financial Damage

3 - Minor effect on annual profit

Reputation Damage

3

Non-compliance

3

Privacy Violation

1

Business Impact Factor: Low (BIF: 2.5)

Impact Factor: Low (IF: 2.5)

Overall Risk Severity: Note

Score Vector: (SL:3/M:2/O:2/S:1/ED:3/EE:3/A:5/ID:4/LC:1/LI:2/LAV:2/LAC:3/FD:3/RD:3/NC:3/PV:1)

Recomendaciones:

- No activar las funciones de depuración en servidores de producción ni en aplicaciones destinadas a los usuarios finales.

Para el código que se relacione con flask se debe colocar lo siguiente:

```
from flask import Flask

app = Flask()
app.debug = False
app.run(debug=False)
```

Problema 4

0.0% Security Hotspots Reviewed

To review

Fixed

Safe

4 Security Hotspots to review

Review priority: High

Cross-Site Request Forgery (CSRF)

1

>

Review priority: Medium

Permission

1

>

Review priority: Low

Insecure Configuration

1

>

Make sure this debug feature is deactivated before delivering the code in production.

Others

1

>

Make sure automatically installing recommended packages is safe here.

4 of 4 shown

Make sure automatically installing recommended packages is safe here.

Automatically installing recommended packages is security-sensitive [docker:S8500](#)

Status: To Review

This Security Hotspot needs to be reviewed to assess whether the code poses a risk.

Where is the risk?

What's the risk?

Assess the risk

How can I fix it?

Activity

/Dockerfile

Show 2 more lines

3

4

5

6

7

8

9

10

11

12

13

Show 8 more lines

Figura 5. Resultado de problema de seguridad nivel bajo en asuntos de "otros"

Identificador	Problema de seguridad WT-4
Título	Instalación y revisión de paquetes
Descripción	Instalar paquetes innecesarios en imágenes Docker aumenta la superficie de ataque, pudiendo contener vulnerabilidades o código malicioso. Esto puede facilitar ataques a la cadena de suministro o escaladas de privilegios.
Referencias	<ul style="list-style-type: none"> OWASP_2021_A06 CWE-937
Riesgo	<p>OWASP Risk Rating Calculator</p> <p>Likelihood Factors</p> <p>Threat Agent Factors Skill Level: 1 - Security penetration skills</p> <p>Motive: 2</p> <p>Opportunity: 1</p> <p>Size: 2 - Developers or system administrators</p> <p>Threat Agent Factor: Low (TAF: 1.5)</p> <p>Vulnerability Factors Ease of Discovery: 2</p> <p>Ease of Exploit: 2</p> <p>Awareness: 4 - Hidden</p> <p>Intrusion Detection: 4</p> <p>Vulnerability Factor: Medium (VF: 3)</p> <p>Impact Factors</p> <p>Technical Impact Factors Loss of Confidentiality: 1</p> <p>Loss of Integrity: 1 - Minimal slightly corrupt data</p> <p>Loss of Availability: 1 - Minimal secondary services interrupt</p> <p>Loss of Accountability: 2</p> <p>Technical Impact Factor: Low (TIF: 1.25)</p> <p>Business Impact Factors Financial Damage: 2</p> <p>Reputation Damage: 1 - Minimal damage</p> <p>Non-compliance: 2 - Minor violation</p> <p>Privacy Violation: 1</p> <p>Business Impact Factor: Low (BIF: 1.5)</p> <p>Likelihood Factor: Low (LF: 2.25)</p> <p>Impact Factor: Low (IF: 1.5)</p> <p>Overall Risk Severity: Note</p> <p>Score Vector: (SL:1/M:2/O:1/S:2/ED:2/EE:2/A:4/ID:4/LC:1/LI:1/LAV:1/LAC:2/FD:2/RD:1/NC:2/PV:1)</p>

Recomendaciones:

- Evitar la instalación de paquetes innecesarios.
- Verificar si existen vulnerabilidades en los paquetes requeridos.

Si nos vamos por la parte de comandos pues lo recomendable sería usar lo siguiente:

RUN apt update && apt --no-install-recommends install -y build-essential && rm -rf /var/lib/apt/lists/*

- apt update** → Evita instalar paquetes obsoletos o vulnerables.
- no-install-recommends** → Reduce dependencias innecesarias, limitando la superficie de ataque.
- rm -rf /var/lib/apt/lists/*** → Elimina caché de apt, reduciendo el tamaño y ocultando información sensible.