



ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

Desarrollo de Software Seguro

Análisis estático del código

Kelly Sangoluisa



Contenido

1. Herramientas usadas	3
2. Resumen ejecutivo	3
3. Hallazgos	4
Hallazgo 1	4
Hallazgo 2	5
Hallazgo 3	6
Hallazgo 4	8
Hallazgo 5	9
4. Anexos	10



1. Herramientas usadas

Para el siguiente análisis se usó algunas herramientas como:

- **Bandit**, la cual es una herramienta de análisis estático especializada en el lenguaje Python, el lenguaje principal utilizado en el examen.
- **SonarCloud**, la usamos para complementar los resultados de Bandit y además de su uso para integrarla a GitHub y así automatizar los nuevos cambios realizados durante el examen.

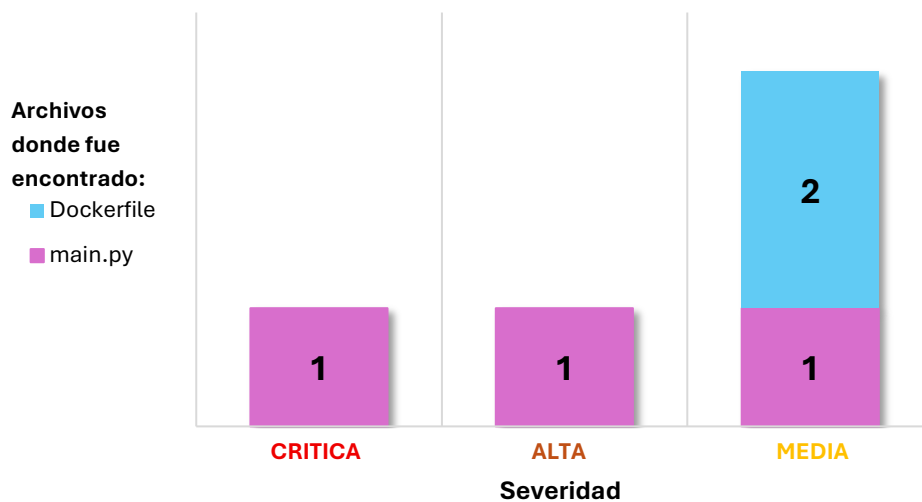
2. Resumen ejecutivo

Mediante un análisis manual y con ayuda de las herramientas mencionadas anteriormente, se encontraron:

- 5 vulnerabilidades de seguridad.
- 1 de severidad crítica, 1 de severidad alta y 3 severidad de media.

Además, estas se encuentran solo en dos archivos, *main.py* y *Dockerfile*.

TOTAL DE HALLAZGOS



Todas las vulnerabilidades encontradas en el archivo *main.py* han sido mitigadas, y las encontradas en el *Dockerfile* por el momento no, debido a las instrucciones del examen, pero se encuentra una recomendación correspondiente en el caso de implementar la mitigación.



3. Hallazgos

Hallazgo 1

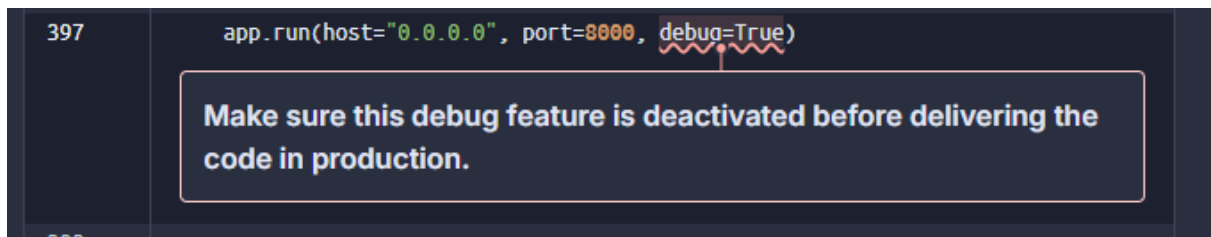
Código de identificación del hallazgo: H01

Descripción de la vulnerabilidad:

Se encontró un `debug=true`, lo que significa que está habilitado el depurador, y en el caso de acceder desde el navegador se podría ingresar comandos peligrosos para todo el sistema y sus datos. lo cual lo hace una vulnerabilidad grave de seguridad si se manda así a producción.

Origen:

Archivo: .\app\main.py Línea: 397



Severidad:

Con la herramienta de CVSS el puntaje obtenido es 9.8 por lo que es una Severidad Crítica

Base Score

9.8
(Critical)

Attack Vector (AV)	Scope (S)
<input checked="" type="radio"/> Network (N) <input type="radio"/> Adjacent (A) <input type="radio"/> Local (L) <input type="radio"/> Physical (P)	<input checked="" type="radio"/> Unchanged (U) <input type="radio"/> Changed (C)
Attack Complexity (AC)	Confidentiality (C)
<input checked="" type="radio"/> Low (L) <input type="radio"/> High (H)	<input type="radio"/> None (N) <input type="radio"/> Low (L) <input checked="" type="radio"/> High (H)
Privileges Required (PR)	Integrity (I)
<input checked="" type="radio"/> None (N) <input type="radio"/> Low (L) <input type="radio"/> High (H)	<input type="radio"/> None (N) <input type="radio"/> Low (L) <input checked="" type="radio"/> High (H)
User Interaction (UI)	Availability (A)
<input checked="" type="radio"/> None (N) <input type="radio"/> Required (R)	<input type="radio"/> None (N) <input type="radio"/> Low (L) <input checked="" type="radio"/> High (H)

Vector String -
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H



Recomendación / Mecanismo de mitigación:

Aunque esto se puede usar en desarrollo, para evitar problemas como los detectados, se recomienda cambiarla a `debug=False`.

```
1 file changed +1 -1 lines changed
core-bankek-python/app/main.py
@@ -610,4 +610,4 @@ def initialize_db():
610     g.db_initialized = True
611
612     if __name__ == "__main__":
613 - app.run(host="0.0.0.0", port=8000, debug=True)
613 + app.run(host="127.0.0.1", port=8000, debug=False)
```

Hallazgo 2

Código de identificación del hallazgo: H02

Descripción de la vulnerabilidad:

Se encontró que la aplicación está usando un `(host="0.0.0.0")`, lo cual hace que se acepten conexiones desde cualquier IP, aumentando los riesgos de vulnerabilidad si no se tiene un control adecuado de acceso o se aplica un firewall.

Origen:

Archivo: .app\main.py Línea: 397

```
396 if __name__ == "__main__":
397     app.run(host="0.0.0.0", port=8000, debug=True)
398
```

Severidad:

Con la herramienta de CVSS el puntaje obtenido es 6.5 por lo que es una Severidad Media

Base Score

6.5 (Medium)

Attack Vector (AV): Network (N) [Selected], Adjacent (A), Local (L), Physical (P)

Attack Complexity (AC): Low (L) [Selected], High (H)

Privileges Required (PR): None (N) [Selected], Low (L), High (H)

User Interaction (UI): None (N) [Selected], Required (R)

Scope (S): Unchanged (U) [Selected], Changed (C)

Confidentiality (C): None (N), Low (L) [Selected], High (H)

Integrity (I): None (N), Low (L) [Selected], High (H)

Availability (A): None (N), Low (L), High (H)

Vector String -
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N



Recomendación / Mecanismo de mitigación:

Se debe proteger el acceso con firewalls o proxy inversos, o su vez usar interfaces específicas como "host="127.0.0.1" en caso de desarrollo y para cuando este en producción usar una ip específica.

```
@@ -610,4 +610,4 @@ def initialize_db():
610     g.db_initialized = True
611
612     if __name__ == "__main__":
613 -         app.run(host="0.0.0.0", port=8000, debug=True)
613 +         app.run(host="127.0.0.1", port=8000, debug=False)
```

Hallazgo 3

Código de identificación del hallazgo: H03

Descripción de la vulnerabilidad:

Se encontró que la aplicación flask es sensible a ataques CSRF, ya que no se encuentra alguna configuración para controlarla, esto es una vulnerabilidad grande ya que se pueden realizar solicitudes peligrosas en nombre de algún usuario autenticado, poniendo en riesgo la confidencialidad e integridad de sus datos.

Origen:

Archivo: ./app/main.py Línea 32

```
31
32     app = Flask(__name__)
33     api = Api(
```

Severidad:

Con la herramienta de CVSS el puntaje obtenido es 7.3 por lo que es una Severidad Alta

Base Score: 7.3 (High)

Attack Vector (AV): Network (N) [Selected], Adjacent (A), Local (L), Physical (P)

Attack Complexity (AC): Low (L) [Selected], High (H)

Privileges Required (PR): None (N), Low (L) [Selected], High (H)

User Interaction (UI): None (N), Required (R) [Selected]

Scope (S): Unchanged (U) [Selected], Changed (C)

Confidentiality (C): None (N), Low (L), High (H) [Selected]

Integrity (I): None (N), Low (L), High (H) [Selected]

Availability (A): None (N) [Selected], Low (L), High (H)

Vector String -
CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:N



Recomendación / Mecanismo de mitigación:

Se recomienda habilitar la protección CSRF en la aplicación, o si es el caso configurarla.

En el examen como mecanismo de mitigación se usará verificación por medio de JWT para todos los endpoints, primero se envía el header *Authorization* en lugar de almacenarse en cookies y así evitar CSRF:

```
85  v authorizations = {
86  v    'Bearer': {
87      'type': 'apiKey',
88      'in': 'header',
89      'name': 'Authorization',
90      'description': "Enter your token in the format **Bearer <token>**"
91    }
92  }
```

Luego el decorator asegura que el token este en el header *Authorization*, y validamos el token:

```
121  @wraps(f)
122  def decorated(*args, **kwargs):
123      # Obtener el header de autorización
124      auth_header = request.headers.get("Authorization", "")
125
126      if not auth_header.startswith("Bearer "):
127          abort(401, "Authorization header missing or invalid")
128
129      # Extraer el token
130      token = auth_header.split(" ")[1]
131
132      # Obtener el gestor JWT desde el contexto de la aplicación
133      jwt_manager = current_app.jwt_manager
134
135      # Verificar el token
136      payload = jwt_manager.verify_token(token)
137      if not payload:
138          abort(401, "Invalid or expired token")
139
```



Hallazgo 4

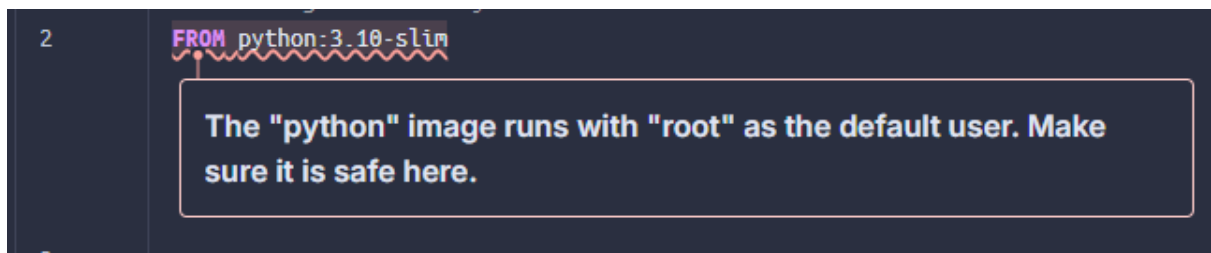
Código de identificación del hallazgo: H04

Descripción de la vulnerabilidad:

En la imagen de Docker se encontró que python:3.10-slim ejecuta los procesos dentro del contenedor con el usuario root por defecto, siendo una vulnerabilidad por los privilegios que se obtienen con el usuario root, ya que si se tiene acceso a este se compromete el sistema.

Origen:

Archivo: ./ Dockerfile Línea 2



Severidad:

Con la herramienta de CVSS el puntaje obtenido es 6.3 por lo que es una Severidad Media

Base Score		6.3 (Medium)
Attack Vector (AV)	Scope (S)	
<input type="button" value="Network (N)"/> <input checked="" type="button" value="Adjacent (A)"/> <input type="button" value="Local (L)"/> <input type="button" value="Physical (P)"/>	<input checked="" type="button" value="Unchanged (U)"/> <input type="button" value="Changed (C)"/>	
Attack Complexity (AC)	Confidentiality (C)	
<input checked="" type="button" value="Low (L)"/> <input type="button" value="High (H)"/>	<input type="button" value="None (N)"/> <input checked="" type="button" value="Low (L)"/> <input type="button" value="High (H)"/>	
Privileges Required (PR)	Integrity (I)	
<input checked="" type="button" value="None (N)"/> <input type="button" value="Low (L)"/> <input type="button" value="High (H)"/>	<input type="button" value="None (N)"/> <input checked="" type="button" value="Low (L)"/> <input type="button" value="High (H)"/>	
User Interaction (UI)	Availability (A)	
<input checked="" type="button" value="None (N)"/> <input type="button" value="Required (R)"/>	<input type="button" value="None (N)"/> <input checked="" type="button" value="Low (L)"/> <input type="button" value="High (H)"/>	
Vector String - CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L		

Recomendación / Mecanismo de mitigación:

Crear y configurar un usuario, con los permisos necesarios, no con acceso grant como el root, para mitigar el riesgo de la obtención maliciosa del usuario.



Hallazgo 5

Código de identificación del hallazgo: H05

Descripción de la vulnerabilidad:

En el archivo Docker se encontró que el comando para instalar paquetes, *apt-get install -y gcc libpq-dev*, también instala paquetes recomendados, y no solamente los necesarios, lo cual puede convertirse en una vulnerabilidad ya que al aumentar paquetes recomendados de los cuales no sabemos si son seguros aumentar el riesgo de ataques por ese medio.

Origen:

Archivo: ./ Dockerfile Linea 8

```
7 # Instalar dependencias del sistema (gcc, libpq-dev para compilar psycopg2)
8 RUN apt-get update && apt-get install -y
  gcc libpq-dev && rm -rf /var/lib/apt/lists/*
```

Make sure automatically installing recommended packages is safe here.

Severidad:

Con la herramienta de CVSS el puntaje obtenido es 5.1 por lo que es una Severidad Media

Base Score

5.1
(Medium)

Attack Vector (AV)	Scope (S)
Network (N) Adjacent (A) Local (L) Physical (P)	Unchanged (U) Changed (C)
Attack Complexity (AC)	Confidentiality (C)
Low (L) High (H)	None (N) Low (L) High (H)
Privileges Required (PR)	Integrity (I)
None (N) Low (L) High (H)	None (N) Low (L) High (H)
User Interaction (UI)	Availability (A)
None (N) Required (R)	None (N) Low (L) High (H)

Vector String -
CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N

Recomendación / Mecanismo de mitigación:

Agregar el comando *--no-install-recommends* para que evitar que se instalen paquetes innecesarios.



4. Anexos

- SonarCloud: <https://sonarcloud.io/>
- CVSS Calculator: <https://www.first.org/cvss/calculator/3-1>
- Bandit: <https://bandit.readthedocs.io/en/latest>