COMO ANALIZAR UN ARCHIVO .PY CON FORTIFY

Fortify ofrece un conjunto de herramientas para analizar la seguridad de las aplicaciones. En el caso específico de un archivo .py (un archivo de código fuente en Python), puede utilizar la herramienta "Fortify Static Code Analyzer" (SCA) para analizar su código en busca de vulnerabilidades.

Para analizar un archivo .py con Fortify SCA, siga estos pasos:

1. Abra Fortify SCA y seleccione la opción "New Analysis" (Nuevo análisis) en el menú "File" (Archivo).
2. Seleccione "Python" en la lista de lenguajes admitidos y haga clic en "Next" (Siguiente).
3. En la ventana de configuración del análisis, seleccione la ubicación del archivo .py que desea analizar y configure las opciones de análisis según sea necesario.
4. Haga clic en "Finish" (Finalizar) para iniciar el análisis. Fortify SCA analizará el código fuente del archivo .py en busca de vulnerabilidades de seguridad y generará un informe de resultados.

Es importante tener en cuenta que Fortify SCA requiere una licencia para su uso. Además, es recomendable realizar análisis de seguridad en el código fuente de forma regular durante el ciclo de vida del desarrollo de software para detectar y solucionar vulnerabilidades antes de que se conviertan en problemas de seguridad en producción.

EJEMPLO DE PRUEBA EN FORTIFY

Aquí tienes un código de ejemplo en Python que contiene algunas vulnerabilidades comunes como inyección de SQL, uso de eval, uso de exec y uso de pickle. Podrías usar Fortify para escanear este código y generar un reporte interesante con las recomendaciones para corregir estas vulnerabilidades.

```
import sqlite3
import sys
import pickle

conn = sqlite3.connect('example.db')
c = conn.cursor()

# Vulnerable a inyección de SQL
user_input = input("Enter your name: ")
c.execute("SELECT * FROM users WHERE name = '%s'" % user_input)

# Uso de eval
user_input = input("Enter a math expression: ")
result = eval(user_input)
print("The result is", result)

# Uso de exec
user_input = input("Enter some Python code: ")
exec(user_input)

# Uso de pickle
user_input = input("Enter a file name: ")
with open(user_input, "rb") as f:
    data = pickle.load(f)
    print("The data is", data)
```

```
conn.close()
```