

Universidad del Valle de Guatemala
Seguridad en Sistemas de Computación
Elean Rivas - 19062
Diego Ruiz - 18761
José Jorge Pérez - 18364
Kenneth Gálvez - 20079

Seguridad en operaciones y en desarrollo | SonarCloud

Con el objetivo de aprender más sobre el análisis de código automatizado y sus beneficios en el desarrollo de software, se realizó esta hoja de trabajo. Para ello, se conectó SonarCloud a un repositorio de GitHub que contenía un ejemplo de código en Python. Este código, a propósito, tenía varios errores y vulnerabilidades que debían ser identificados y corregidos. SonarCloud, una herramienta avanzada de análisis estático de código, fue utilizada para escanear el repositorio y detectar automáticamente estos problemas, ofreciendo reportes detallados y sugerencias para su resolución. Este proceso permitió una comprensión más profunda de cómo las herramientas de análisis de código pueden mejorar la calidad del software, aumentar la eficiencia en la detección de errores y contribuir a la creación de proyectos más robustos y mantenibles.

Los errores que encontramos en el código provienen principalmente de problemas comunes que pueden afectar gravemente tanto la funcionalidad como la mantenibilidad de una aplicación Python. Una preocupación importante es la gestión de recursos, específicamente cómo se manejan los archivos dentro de las funciones `read_file` y `write_file`. En ambos casos, los archivos se abren de manera tradicional sin utilizar el administrador de contexto proporcionado por la declaración `'with'` de Python. Este enfoque puede llevar a que los recursos no se liberen correctamente si ocurre una excepción antes de que el archivo se cierre manualmente, lo que no solo es un error típico de principiante, sino también una fuente potencial de corrupción de archivos y pérdida de datos en escenarios más severos.

Los errores encontrados en el código provienen principalmente de problemas comunes que pueden afectar gravemente tanto la funcionalidad como la mantenibilidad de una aplicación Python. Un problema significativo es la gestión de recursos, específicamente cómo se manejan los archivos dentro de las funciones `'read_file'` y `'write_file'`. En ambos casos, los archivos se abren de manera tradicional sin utilizar el administrador de contexto proporcionado por la declaración `'with'` de Python. Este enfoque puede llevar a que los recursos no se liberen correctamente si ocurre una excepción antes de que el archivo se cierre manualmente. Esto no solo es un error típico de principiante, sino también una fuente potencial de corrupción de archivos y pérdida de datos en escenarios más severos.

Otro problema significativo identificado en el código está relacionado con un error de tipo dentro de la función `process_data`. En este caso, el código intenta concatenar una cadena con un entero, lo cual Python prohíbe estrictamente. Este error conduce a un `TypeError`, que podría detener la ejecución del programa inesperadamente durante el tiempo de ejecución. Tales errores de tipo indican una falta de validación de datos adecuada o de verificación de tipos dentro del código, reflejando un problema más amplio en cuanto a garantizar que las funciones manejen de manera robusta los tipos de datos que se espera que procesen.

Tras realizar una exploración más exhaustiva, el problema de la falta de coincidencia de tipos no se trata solo de un solo error; revela una falta más profunda de programación defensiva. La función asume que la entrada siempre se ajustará a los tipos esperados sin realizar ninguna verificación. Esta suposición puede llevar a vulnerabilidades, especialmente en aplicaciones que interactúan con entradas externas o APIs donde los tipos de datos y su integridad no siempre pueden garantizarse.

Además, estos errores también pueden implicar una falta de pruebas exhaustivas o la ausencia de un proceso de revisión de código minucioso. Las pruebas efectivas, especialmente las pruebas unitarias, idealmente deberían detectar tales errores antes de que el código se implemente en producción. La ausencia de estas pruebas sugiere una posible omisión en el proceso de desarrollo o un malentendido de las mejores prácticas en el desarrollo de software, especialmente en un lenguaje de tipado dinámico como Python donde tales errores no se capturan en tiempo de compilación.

Además, estos problemas podrían ser sintomáticos de desafíos más amplios dentro del entorno de desarrollo o del equipo. Por ejemplo, si errores básicos como estos son frecuentes, podría sugerir la necesidad de mejorar la capacitación de los desarrolladores o de establecer mejores estándares de codificación. Invertir en estas áreas no solo podría mitigar errores similares en el futuro, sino también mejorar la calidad general y la seguridad del software que se está desarrollando.

En esencia, aunque los errores específicos como el mal uso y fuga de recursos y la probabilidad de caer en problemas de tipo en el runtime de la aplicación son técnicos en naturaleza, sus implicaciones son de gran alcance, afectando aspectos de seguridad, confiabilidad y mantenibilidad del código. Abordar estos problemas requiere no sólo correcciones técnicas, sino que podría representar un cambio significativo en términos de ajustes organizativos o procedimentales más amplios para fomentar una cultura de calidad y precisión en las prácticas de programación. Un enfoque integral garantiza no solo la corrección de problemas actuales, sino también la prevención de problemas similares en desarrollos futuros.

S

sonarcloud-analysis

No tags
Last analysis 22 May 2024 94 Lines of Code


XML

Python

Main Branch Status

Quality Gate ?

Passed



Enjoy your sparkling clean code!

See Full Analysis

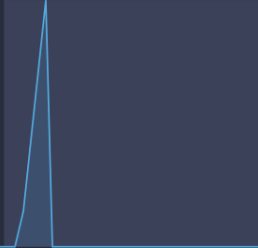
Main Branch Evolution since 3 hours ago

0 Issues

Issues

Coverage

Duplications



Issues

New Code

See full history

Latest Activity

NEW ANALYSIS

Main Branch

22 May at 12:45

c1cc1065

aaaaaaaa

Passed

0 Fixed Issues

0 New Issues

0.0% Coverage

0.0% Duplications

0 Lines of Code

Show Older Activity

ah18051 / sonarcloud-analysis

CodeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

Actions

New workflow

All workflows

SonarCloud

Management

Caches

Alerts

Runners

All workflows

Showing runs from all workflows

Filter workflow runs

26 workflow runs

Event	Status	Branch	Author
aaaaaaa	Success	main	ah18051
Delete pruebapy	Success	main	ah18051
si	Success	main	ah18051
Create pruebapy	Success	main	ah18051
Update sonar-project.properties	Success	main	ah18051
Update sonar-project.properties	Success	main	ah18051