Universidad del Valle de Guatemala
Facultad de ingeniería

Proyecto Análisis Sonarcloud

Cayetano Molina 20211
Estefanía Elvira 20725
Priscilla Gonzalez 20689

Guatemala 22 de mayo del 20224

Resultados del análisis

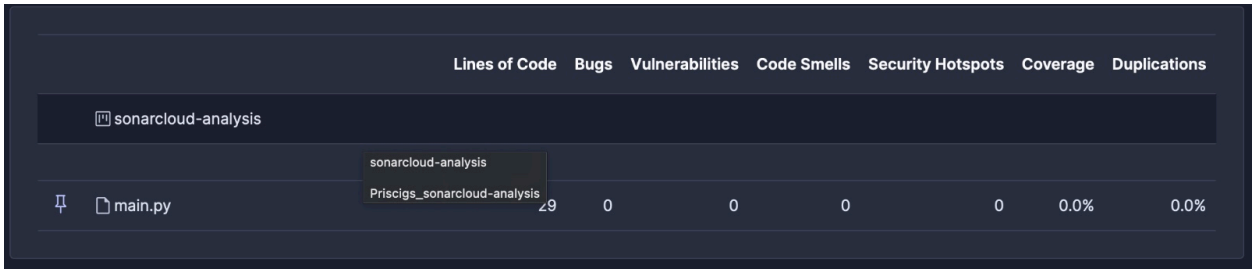

Imagen no.1: Resultados del código sin ninguna vulnerabilidad

Imagen no. 2: Resultados con Code Smells y Security Hostpots

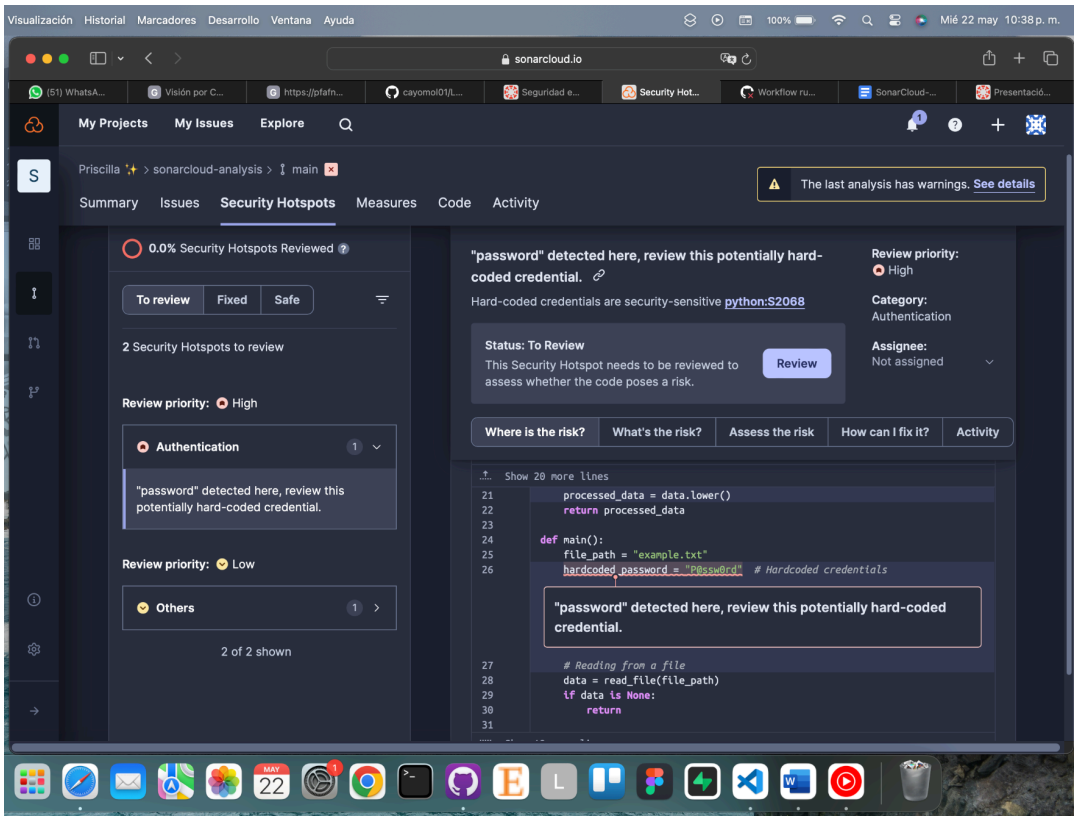

Imagen no. 3: Main branch

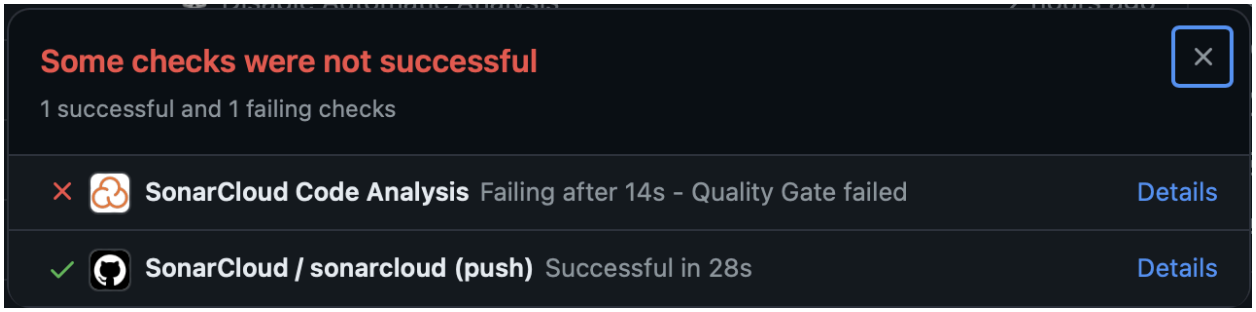

Imagen no. 4: Referencia desde Github

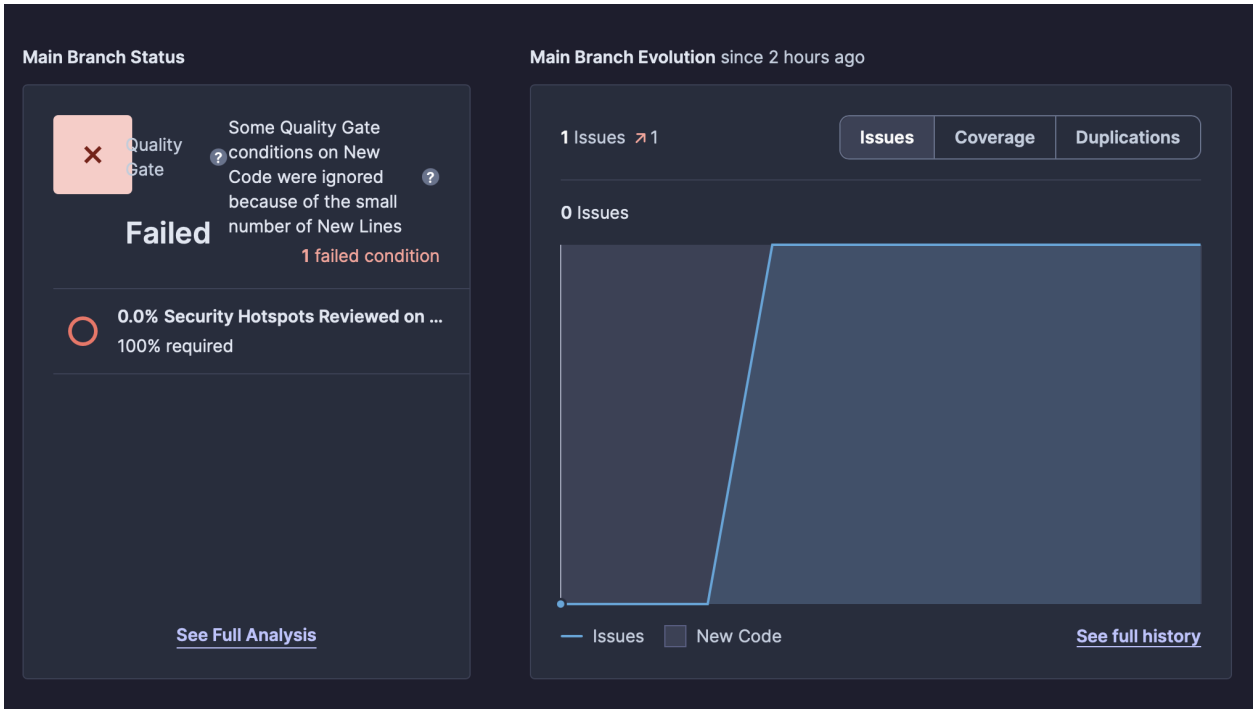

Imagen no. 5: Overview

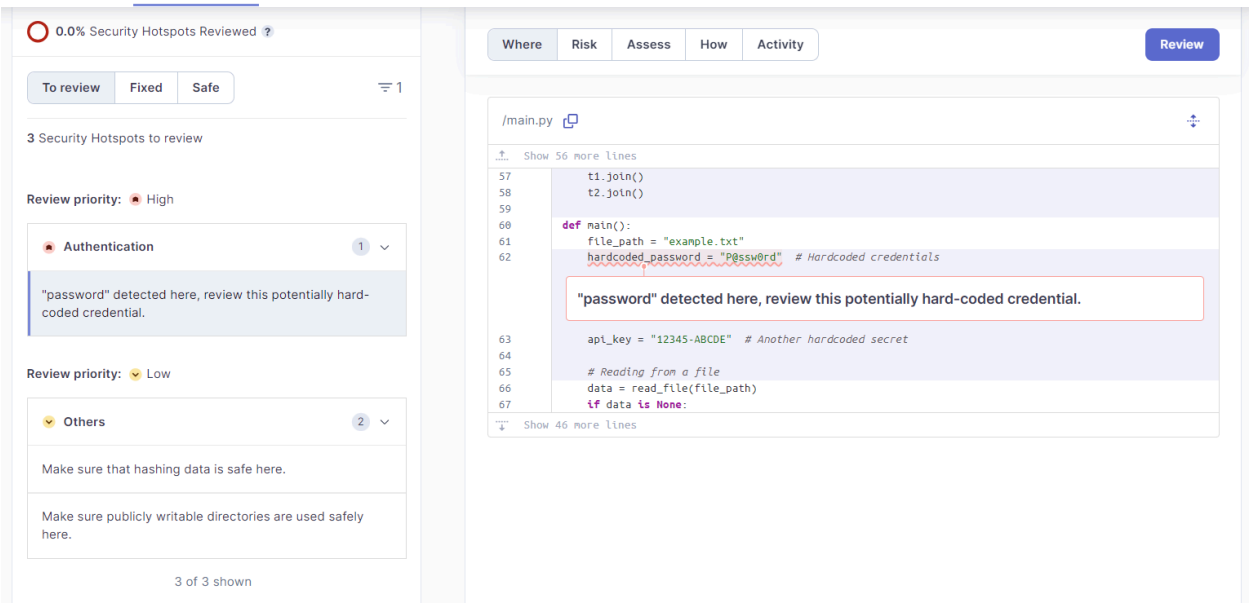

Imagen no. 6: Problemas en el código

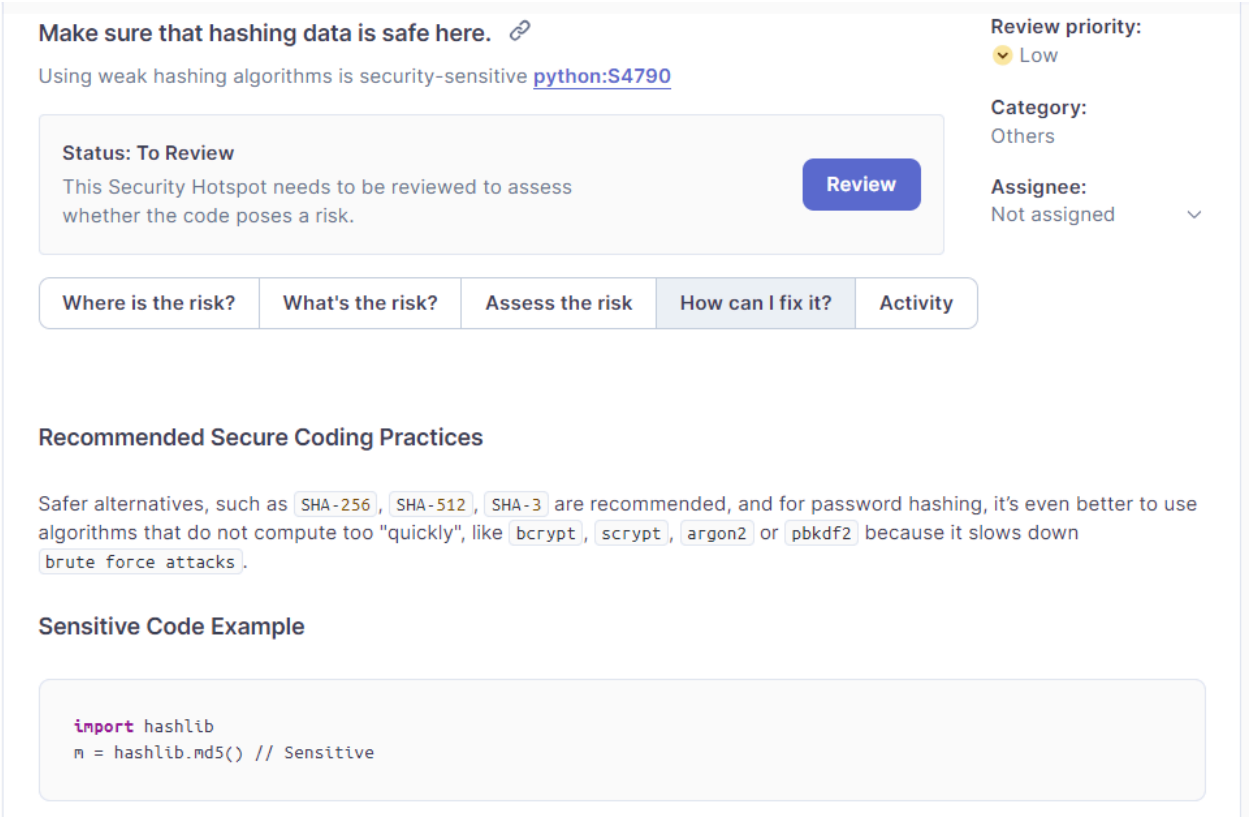

Imagen no. 7: Vulnerabilidades en el código (hashing)

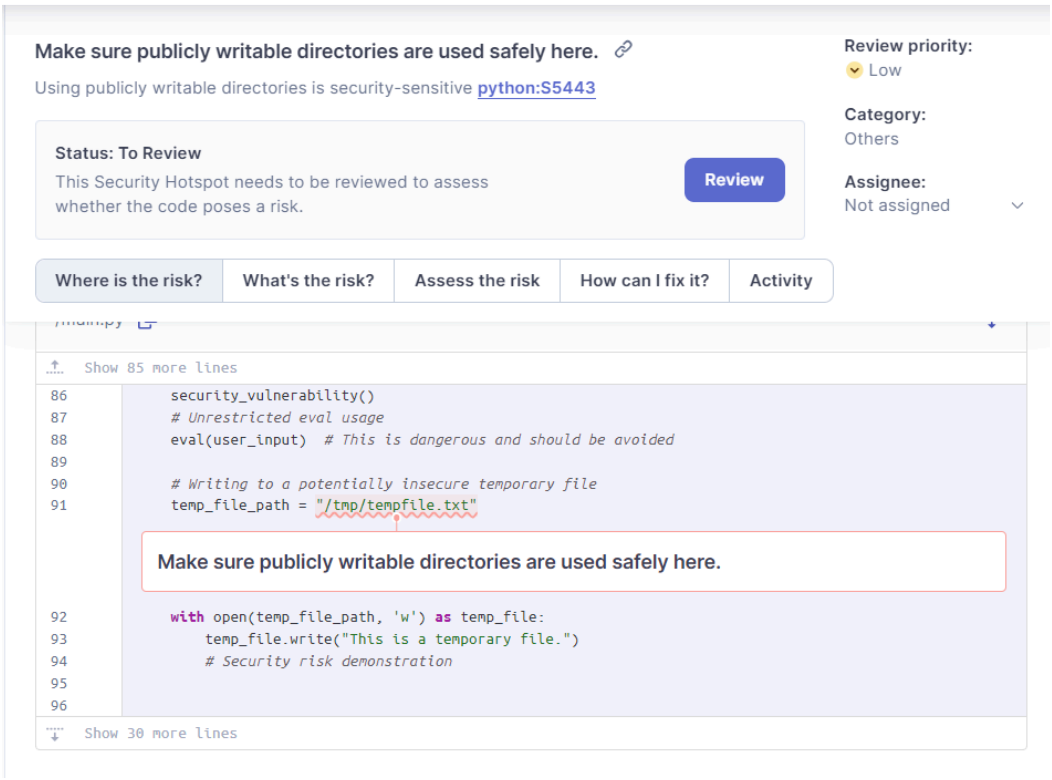

Imagen no. 8: Sensitive code Vulnerabilidad

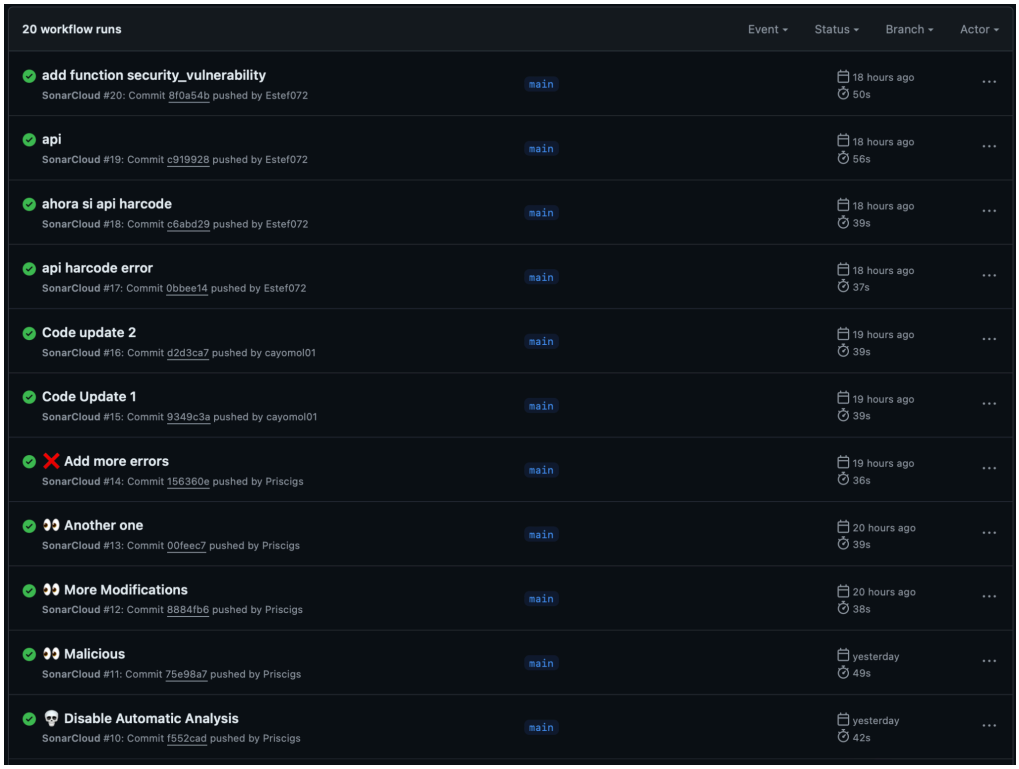

Imagen no. 9: Workflow de Github

Discusión

Se marca un problema grave (mostrado en la imagen 8) en el código el cual dice que es de autenticación. En este problema se está escribiendo una contraseña completamente dentro del código lo cual puede llevar a problemas más adelante. La solución sería utilizar variables de entorno o variables que no se suban al repositorio pues pueden quedar grabadas en algún lado. O bien, si se desea guardar una contraseña, se puede utilizar una función de hashing para solo tener que comprobar los caracteres y no tener que revelar las contraseñas en ningún momento.

Sin embargo, sonarcloud no sólo marca los problemas que podrían ser graves sino que además detecta ciertas vulnerabilidades que podrían ser riesgos en el código. Como se muestra en la imágen 7, se está marcando una vulnerabilidad en la función de hashing utilizada dentro del código. Es bien conocido que un algoritmo de hash tiene que ser bueno para que sea útil, Por lo tanto, si se quiere resolver esa advertencia valdría la pena investigar los mejores algoritmos utilizados actualmente para la función de hashing.

Por otra parte, se observa que la condición del **if = _main_** está escrita de forma incorrecta. Para resolverlo, simplemente se debería de escribir así **if = __main__.** También, se encuentra el eval() que contiene la entrada de usuario que puede llegar a ser bastante peligroso al ejecutar código de forma no restringida. Para poder solucionar esta parte del código, lo más simple que se puede hacer es borrar esa línea del código, ya que no es necesaria en este código. O de igual forma, se podría reemplazar con otro código dependiendo las necesidades que se tengan

Otras de las vulnerabilidades o riesgos que se pueden observar es que se está escribiendo código en un directorio temporal ya que los permisos pueden llegar a afectar. Para solucionar esto, se podría hacer la escritura de código en otro directorio o aplicar restricciones de acceso adecuadas.

Por último, existen en el código variables que se definen, pero nunca se usan. Entonces lo ideal podría ser que se comenten o simplemente se borren las variables que no están siendo utilizadas.